Computer Science Review 39 (2021) 100357

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Review article

Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation

Bayu Adhi Tama [a], Sunghoon Lim [b,*]

[a] Data Science Group, Center for Mathematical and Computational Sciences, Institute for Basic Science (IBS), Daejeon 34126, Republic of Korea
[b] Department of Industrial Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea

ARTICLE INFO

Article history:
Received 7 July 2020
Received in revised form 23 November 2020
Accepted 22 December 2020
Available online 30 December 2020

Keywords:
Intrusion detection systems
Anomaly detection
Ensemble learners
Combination methods
Tree-based classifier ensemble
Stacking
Systematic mapping study
Empirical review

ABSTRACT

Intrusion detection systems (IDSs) are intrinsically linked to a comprehensive solution of cyberattacks prevention instruments. To achieve a higher detection rate, the ability to design an improved detection framework is sought after, particularly when utilizing ensemble learners. Designing an ensemble often lies in two main challenges such as the choice of available base classifiers and combiner methods. This paper performs an overview of how ensemble learners are exploited in IDSs by means of systematic mapping study. We collected and analyzed 124 prominent publications from the existing literature. The selected publications were then mapped into several categories such as years of publications, publication venues, datasets used, ensemble methods, and IDS techniques. Furthermore, this study reports and analyzes an empirical investigation of a new classifier ensemble approach, called stack of ensemble (SoE) for anomaly-based IDS. The SoE is an ensemble classifier that adopts parallel architecture to combine three individual ensemble learners such as random forest, gradient boosting machine, and extreme gradient boosting machine in a homogeneous manner. The performance significance among classification algorithms is statistically examined in terms of their Matthews correlation coefficients, accuracies, false positive rates, and area under ROC curve metrics. Our study fills the gap in current literature concerning an up-to-date systematic mapping study, not to mention an extensive empirical evaluation of the recent advances of ensemble learning techniques applied to IDSs.

Contents

* Corresponding author.
E-mail address: sunghoonlim@unist.ac.kr (S. Lim).

https://doi.org/10.1016/j.cosrev.2020.100357
1574-0137/

1. Introduction

The ensemble of classifiers; which is hereafter mentioned as an ensemble learner, has drawn a lot of interest in cybersecurity research, and in an intrusion detection system (IDS) domain is no exception [1–3]. An IDS deals with the proactive and responsive detection of external aggressors and anomalous operations of the server before they make such a massive destruction. As of today, a variety number of cyberattacks has been in perilous situations, placing some organization's critical infrastructures into risk. A successful attack may lead to difficult consequences such as but not limited to financial loss, operational termination, and confidential information disclosure [4]. Moreover, the larger the organization's network, the bigger the chance for attackers to exploit. The complexity of the network may also give rise to vulnerabilities and other specific threats [5]. Therefore, security mitigation and protection strategies should be considered mandatory [6].

A possible protection mechanism such as intrusion detection is indispensable as it involves preventive action used to get rid of any malignant acts within the computer network. An IDS attempts to detect and to block attacks without human intervention by examining network and file access logs, audit trails, and other security-relevant information within the organization [7]. Depending on the detection objectives, an IDS is primarily categorized into two approaches, i.e. anomaly and misuse (signature)-based detection. The former techniques figure out attacks through examining traffic patterns that have deviations from normal patterns. Hence, one merit is that they are able to locate previously unknown attacks, however, they retain to have high false positive rate (FPR). Quite the contrary, the latter performs attack detection based on some known attack signatures. Utilizing a pattern-matching algorithm, an attack pattern candidate in the network is checked by comparing it with those predetermined signatures. This results a lower FPR, but fails to detect novel attack patterns [8].

An ensemble learner is built upon several trainable classifiers, e.g. base learners. Each base learner is trained and performs prediction for a particular class label, where final prediction is made using a particular blending technique, e.g. a combiner. In the purview of IDS, vast majority studies on combining classifiers have been initially begun with a single rationale, however, it can be assumed that since then the classifier ensembles perform better than an individual classifier because of several justifications, i.e. statistical, computational, and representational reasons [9]. Impressed with such rationales, this paper exploits the use of state-of-the-art ensemble learners in IDSs through a systematic mapping study. Furthermore, it extends to carry out an empirical benchmark of different combiner techniques, providing researchers a perception and knowledge about the present circumstances and future orientations of ensemble learning applied for IDSs.

The remainder of this paper is comprised of several sections. Section 2 briefly describes the concept of IDSs and ensemble learning. Next, in Section 3, the rationale of this review is explained. This section is followed by Section 4 that outlines the procedure of systematic mapping study. The result of this mapping study is presented in Section 5, while the result of the empirical benchmark is provided in Section 6. Finally, Section 7 summarizes the threat to validity, while Section 8 concludes the paper.

2. Background

This section conveys a big picture of IDSs and ensemble learning.

2.1. Intrusion detection systems

As mentioned earlier, an IDS attempts to monitoring the organization's network infrastructure by detecting the malicious activities in a responsive manner. Liao et al. [10] provide a taxonomy of IDSs with respect to four main different dimensions, i.e. system deployment, timeliness, detection strategy, and data source. Specifically, concerning their deployment strategy, IDSs can be categorized into two technology types, i.e. host-based and network-based. The aim of host-based IDSs (HIDS) is to monitor the occurrences that arise in a local computer system and then to give notification about the findings. One example found in HIDS is the hash of the file system. Any untrust behavior is recognized after comparing the differences between the hash value that is currently recalculated and the one formerly saved in the database. Network-based IDSs (NIDS), on the contrary, are designed to monitor network traffic and to detect malicious activities within the network by examining inflowing network packets.

Besides, concerning the timeliness, IDSs can be deployed in offline or online mode, while anomaly or misuse are the two classifications of IDSs in terms of detection method. An IDS can also be categorized w.r.t the data source obtained for the analysis. This includes how the data is collected, types of data, and where the data is acquired from. Concerning types of data, for instance, the classifications can be network traffic logs, server logs, or application logs. Fig. 1 illustrates the proposed typology of IDSs introduced by Liao et al. [10].

2.2. Ensemble learning

Ensemble learning is also called committee-based learning or multiple classifier systems. It is made up of several base learners which are prevalently originated from training data by a base learning algorithm [11]. Since the objective of an ensemble learning is to boost weak learners, therefore, base learners are also known as weak learners, which can be neural networks, decision trees, Bayesian classifiers, or other types of learning algorithms. Vast majority of ensemble techniques utilizes a

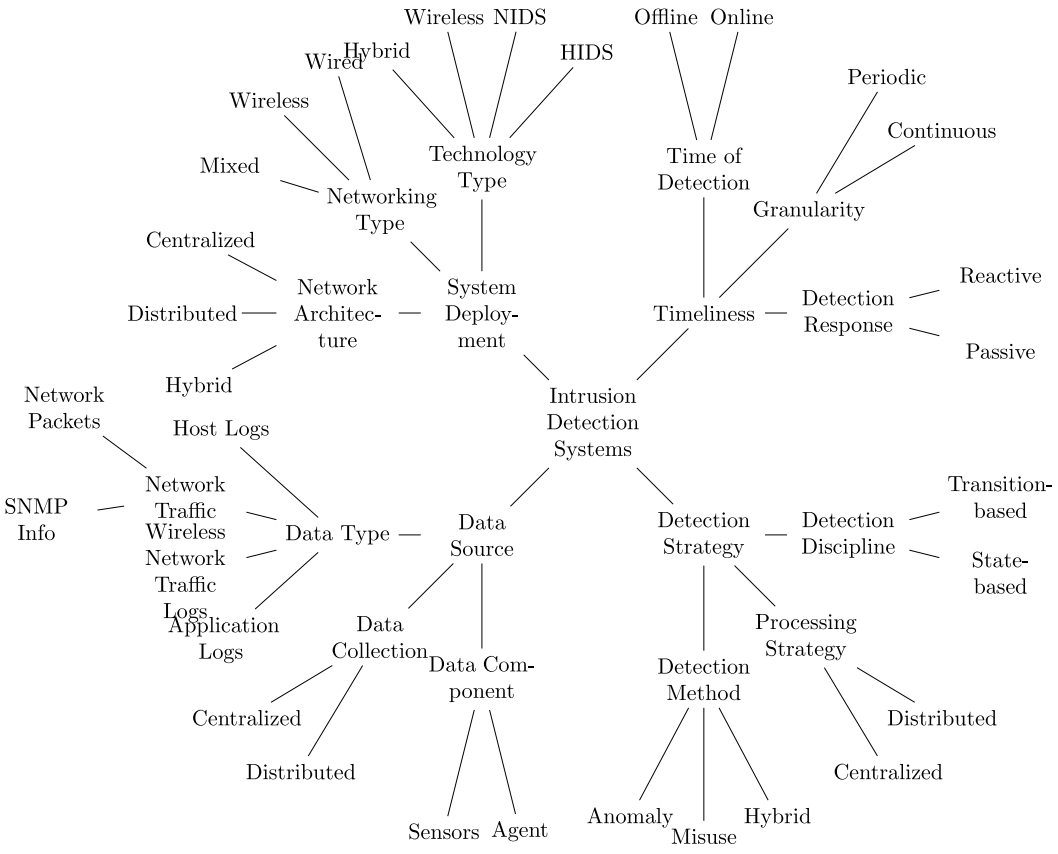

Fig. 1. Typology of intrusion detection systems as discussed in [10].

single base learner to form homogeneous ensemble (e.g., the same kinds of learners), however, there exist particular ensemble techniques which utilize different types of learners, resulting to heterogeneous ensembles.

Kuncheva [12] argues that any ensemble learning can be described in terms of the four level dimensions, i.e. combination level, classifier level, feature level, and data level. For instance, bagging ensembles [13] can be interpreted by answering four sets of questions as follows.

- Combination level
  How are the base learner outputs combined? Voting/average.
- Classifier level
  - Do we utilize same or different base learners? Same base learners.
  - What base learner is best? Decision tree.
  - How many learners are required? 100+
  - Do the M learners be trained together or incrementally? Together.
- Feature level
  Do we utilize all features or use a feature subset for each learner? All feature.
- Data level
  How can we handle the data used for training? Independent bootstrap samples.

Furthermore, Rokach [14] suggests a taxonomy of the ensemble learning (see Fig. 2). The proposed taxonomy possesses five main dimensions, i.e. combiner, diversity, construction of the ensemble, ensemble size, and universality. Likewise, the ensemble learning methodologies can be also be described in terms of the five dimensions.

3. Motivation

Most prior studies have been particularly focused on ensemble learning methods and application architectures. Some survey studies have either emphasize particular ensemble learning [2,3], all-inclusive machine learning algorithms [1,19,21,22], or particular IDSs application architectures [15–18,20]. Moreover, vast majority studies are not derived from a systematic mapping study, making the comprehensiveness and the meaningfulness of the studies remain insignificant. To the best of our knowledge, there are no studies that have reviewed the feasibility of using ensemble learning for IDSs through a systematic mapping study. Besides, there are no review studies that also include an empirical comparison of classifier ensemble methods. Table 1 summarizes the existing survey studies and highlights the importance of our study.

The objective of this study is to bridge the research gap through performing a systematic mapping study. This study emphasizes on the present state of knowledge in ensemble learning-based IDSs. A systematic mapping study was firstly introduced by Petersen et al. [23,24]. It is a research method whose the goals are: (i) to provide an in-depth compendium of area of concern, (ii) to portray the gap of the research, and (iii) to lay down some research notes for future directions. By following this research methodology, we classify ensemble learning-based IDSs, demonstrate how often the publications are, consolidate the results to address some RQs, and show the mapping evidences.

This study cultivates existing bibliographies about the applications of ensemble learning methods in IDSs. We presume that either researchers and practitioners might get the benefit from this study such as designing more advanced and proper IDSs techniques. Even though this is not a panacea for pointing out all the research issues in IDSs; however, it would be a considerable

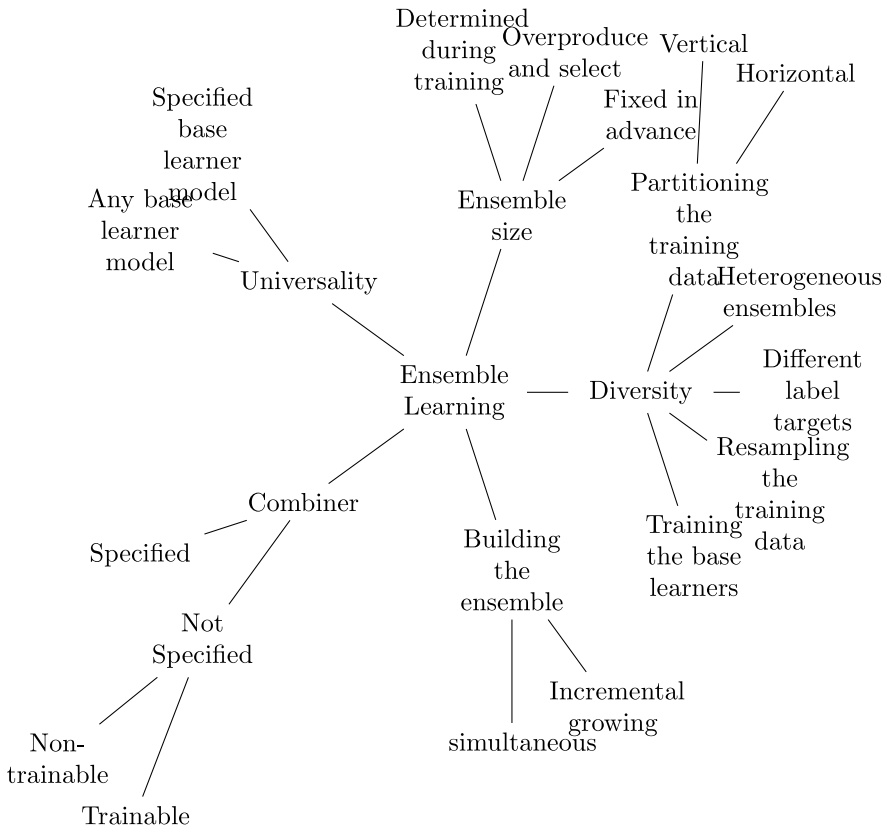

Fig. 2. An ensemble learning taxonomy proposed by Rokach [14].

dawn to develop an advancement in applying ensemble learning for IDSs.

4. Steps in a mapping study

In this section, the practice of carrying out a systematic mapping study is discussed. By following some steps provided in [23–25], we specify some research questions (RQs) being dealt with, search strategy, and requirements for the selection and exclusion.

4.1. Research questions

Kitchenham et al. [25] suggest that RQs should determine the problems being addressed and aim to the research method. We define some following RQs which covers the objective and coverage of our study.

- RQ_1: What is the current trend in ensemble learning-based IDSs?
- RQ_2: What types of ensemble learning methods have been commonly used to cope with the issues arise in IDSs?
- RQ_3: What types of IDS techniques that are developed most?
- RQ_4: What is the relative performance of ensemble learning methods as compared to single classification algorithms?

RQ_1 and RQ_2 are the main research questions, which aim to figure out existing applications of ensemble learning methods in IDSs. By using RQ_3, we aim to identify the most IDSs techniques that were mostly developed. Lastly, RQ_4 aims at cross-benchmarking some ensemble methods over several IDSs datasets. The first-three RQs are addressed in Section 5, while subsequent section (i.e., Section 6) addresses the rest RQ.

4.2. Search strategy

In order to provide an up-to-date review, we consider the studies that were published over the last six years: January 2015 to November 2020. An automatic search is utilized in order to find as many relevant publications as possible. We performed a search from two primary computer science-related digital libraries, i.e. IEEE Digital Library and ACM Digital Library to collect the publications published both in conference proceedings and journals. Besides, we searched the other several widely-acknowledged computing-related digital libraries, i.e. SpringerLink, ScienceDirect, Wiley, Oxford, and Taylor & Francis.

Obtaining good results in carrying out search in those digital libraries needs clearly-defined search keywords. Therefore, we derive some keywords from the above-mentioned RQs and based on keywords identified in some publications. More precisely, the search is on the basis of the fusion of keywords using Boolean operators, i.e. AND and OR, leading to the combined keywords as follows.

> (intrusion detection systems OR anomaly detection OR misuse detection OR signature detection)
> AND
> (classifier ensemble OR ensemble learning OR multiple classifier systems OR voting OR boosting OR bagging OR stacking)

4.3. Requirements for inclusion and exclusion

This section provides the requirements for selection and exclusion specified in our study. The procedure of inclusion and exclusion, as well as the number of studies at each step are summarized in Fig. 3. The searched publications were screened with respect to the following criteria, resulting only the most

Table 1
Summary of previous survey studies in chronological order.

Study	Year	Objective	Demerit	Significance of our study
Folino and Sabatino [15]	2016	State-of-the-art ensemble methods used in IDS are discussed.	Focuses on distributed approaches and implementations.	Empirical benchmark.
Aburomman and Reaz [3]	2017	Various ensemble and hybrid techniques are examined, considering both homogeneous and heterogeneous types of ensemble methods.	The study is not based on a systematic mapping study.	Empirical benchmark and a systematic mapping study.
Sakiz and Sen [16]	2017	Attack detection mechanisms on VANET are surveyed.	Limited to particular application scenarios, e.g. VANET and IoV.	Ensemble learning, a systematic mapping study, and empirical benchmark.
Resende and Drummond [2]	2018	A comprehensive review of the general basic concepts related to IDSs is conducted.	Focuses on particular ensemble techniques, e.g. random forest.	A wide array of ensemble methods are included, a systematic mapping study, and empirical benchmark.
Sultana et al. [17]	2018	The study reviews various recent works on machine learning methods that leverage SDN to implement IDSs.	Emphasizes on SDN-based IDSs.	Ensemble learning, a systematic mapping study, and empirical benchmark.
Chaabouni et al. [18]	2019	The survey classifies the IoT security threats and challenges for IoT networks by evaluating existing defense techniques.	Focuses in IoT-based IDSs.	Ensemble learning, a systematic mapping study, and empirical benchmark.
Chapaneri and Shah [19]	2019	The work surveys the published studies on machine learning-based network IDSs.	The work is not based on a systematic mapping study.	Ensemble learning, a systematic mapping study, and empirical benchmark.
da Costa et al. [20]	2019	Recent and in-depth research of relevant works that deal with several intelligent techniques and their applied IDSs.	Limited to a particular application architecture, i.e. IoT.	Ensemble learning, empirical benchmark, and a systematic mapping study.
Khraisat et al. [1]	2019	The survey aims at providing a taxonomy of IDSs, a comprehensive review of recent works, and an overview of datasets.	Limited number of studies are included.	A systematic mapping study.
Mishra et al. [21]	2019	A detailed investigation and analysis of various machine learning techniques have been conducted in detecting intrusive activities.	The work is not based on systematic mapping study.	A systematic mapping study.
Moustafa et al. [22]	2019	The paper discusses various aspects of anomaly-based IDSs.	The work is not based on a systematic mapping study.	Empirical benchmark, a systematic mapping study.

suitable and relevant studies that are taken into consideration. Existing studies were included based on following criteria.

- I_1: Papers that were published in scholarly venues, i.e. journals, conferences, and workshop proceedings were chosen. These publications had been usually peer-reviewed.
- I_2: Papers that discuss ensemble learning methods for intrusion detection systems were eligible to be selected.

Subsequently, publications that meet at least one of the following requirements were not chosen.

- E_1: Papers discuss ensemble learning methods, but the implementation in IDSs is not discussed. For instance, we excluded Zhu et al. [26] since it is an application of rotation forest for malware detection.
- E_2: Publications that are taken into account as gray literature, i.e. working papers, presentations, and technical reports.
- E_3: Peer-reviewed studies that are not published in journals and proceedings, i.e. PhD thesis and patents.
- E_4: Non-English studies.

5. Results of a mapping study

Driven by the RQs specified in Section 4.1, we defined the following dimensions to map and discuss the selected studies:

- Trend of research: RQ_1
- Publication venues: RQ_1
- The use of datasets: RQ_1
- Types of ensemble schemes: RQ_2
- Types of IDS techniques: RQ_3

The information summarized in Fig. 4, Fig. 5, Table 2, and Table B.8 allows us to answer the first RQ. The remainder parts of this section cover a brief summary of the selected studies that employed ensemble methods in IDSs and categorization of the selected studies w.r.t IDS techniques

5.1. Mapping selected studies by published years

This section provides a research trend in ensemble-based IDSs during the designated period of time. Fig. 4 shows the number of selected studies (i.e., 124 papers) over the specified years, which are from 2015 to 2020. It is obvious that there are at least 15 studies regarding the utilization of ensemble methods for IDSs. According to the trend, there has been a resurgence of interest in implementing the various types of ensemble schemes since 2015. The result indicates that in 2016 and 2020, there have been considerable increase of interest, while on the other hand, there exist only a few number of publications in 2019.

5.2. Mapping selected studies by publication types

The selected studies (i.e., 124 papers) were disseminated as conference papers (i.e., 54 papers), journal papers (i.e., 50 papers), book chapters (i.e., 11 papers), symposium papers (i.e., 6 papers), and workshop papers (i.e., 3 studies). The distribution of selected studies based on publication types is summarized in Fig. 5. Conferences form the predominant part of the publication outlet. The number of ensemble-based IDSs methods that were published in conferences and journals account for approximately

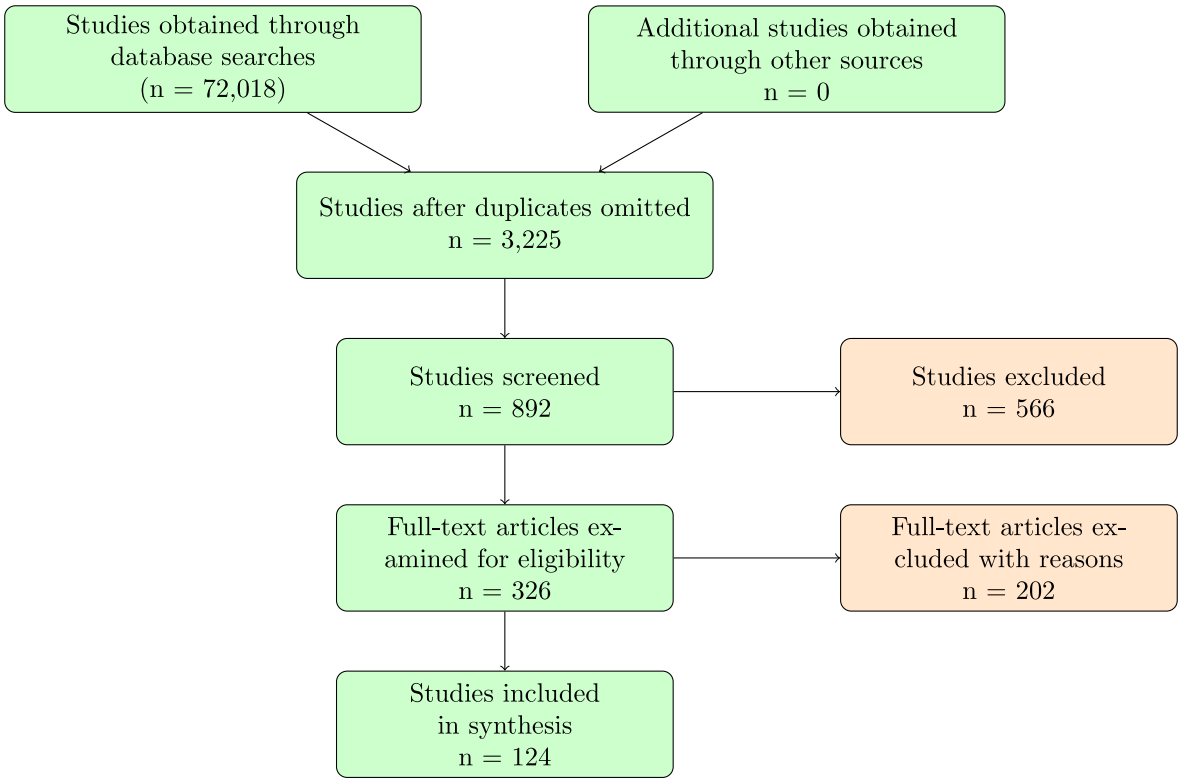

Fig. 3. Flowchart of database search and screening for inclusion and exclusion of existing studies.

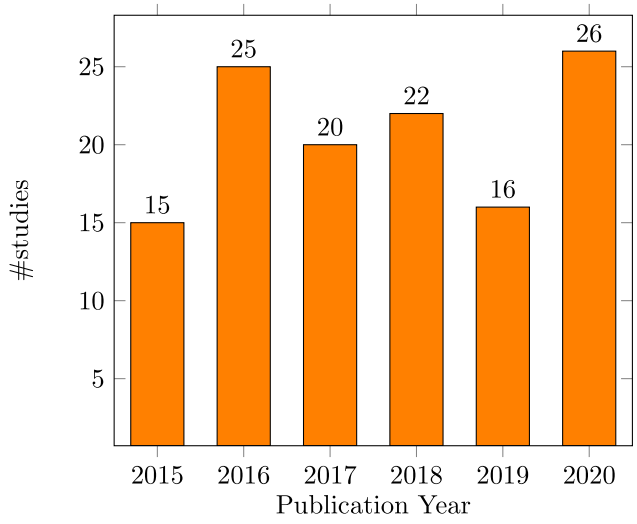

Fig. 4. Distribution of selected studies (i.e., 124 papers) over the period 2015 to 2020.

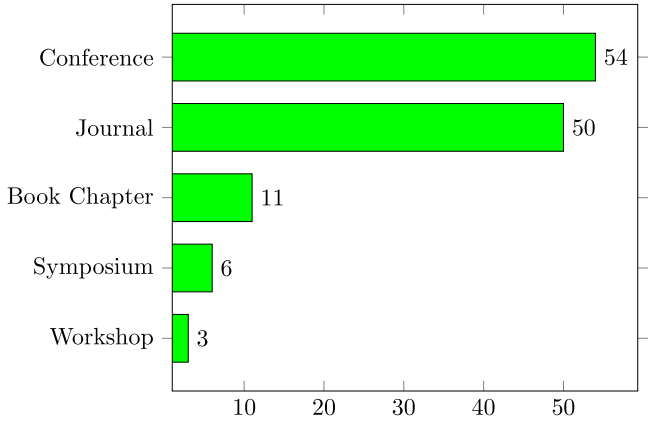

Fig. 5. Distribution of selected studies (i.e., 124 papers) w.r.t publication types.

43% and 40% of the total, respectively. On the contrary, workshop papers were less frequent, which share around 2% of the entire selected studies. Table B.8 in Appendix B enumerates all publication outlets where the selected studies were published in. It presents the outlets with respect to their types, number of studies in each outlet, and the associated percentages.

It is obvious that the selected studies were published in 92 different outlets, where the vast majority of studies were published in IEEE Access (6 publications), Procedia Computer Science (5 publications), IEEE Internet of Things Journal (4 publications), Advances in Intelligent Systems and Computing (3 publications), The Journal of Supercomputing (3 publications), Neural Computing and Applications (3 publications), and Concurrency and Computation: Practice and Experience (3 publications). Other well-known publication outlets are Journal of King Saud University: Computer and Information Sciences, Lecture Notes in Electrical Engineering, Lecture Notes in Computer Science, Journal of Computational Science, Communications in Computer and Information Science, International Conference on Advances in Computing, Communications and Informatics, International Joint Conference on Neural Networks, Computer Networks, Journal of Information Security and Applications, Security and Privacy, Applied Artificial Intelligence, and International Journal of Communication Systems.

Table 2
Summary of selected studies based on IDS datasets considered.

Dataset	Selected studies	#
KDD Cup 99	[27–54]	28
NSL-KDD	[33,36,41,43,46,50,52,55–105]	58
CSIC 2010	[53,102,106,107]	4
Kyoto 2006+	[46,54,80,89,108,109]	6
Wi-Fi Intrusion	[72,88,110,111]	4
Honeypot	[112,113]	2
ISCX 2012	[64,87,89,112,114–116]	7
AWID	[105,117,118]	3
UNSW-NB15	[50,53,72,81,85,88,96,101,102,104,107,119–124]	17
CAIDA	[119,125]	2
CICIDS 2017	[52,53,93,102,105,126,127]	7
DS2OS	[101,128–130]	4
Private	[131–139]	9
Others	LLS-DDoS [125], Contagio [112], UPC [112], ISOT [112], MCFP [112], KPI [140], Telemetry [141], WUIL [142], DARPA 1998 [143], GureKDD [144], GSB [145], Intel Lab [145], Indoor WSN [145], RPL-NDDS17 [146], NIMS [90], UNB-CICT [50], Digiturk [147], Labris [147], IoT Botnet [148], Moore [149], ISCXVPN 2016 [149], CICIDS 2018 [52], Malicious URLs [107], TRAbID [53], CIDDS-001 [104]	25

5.3. Mapping selected studies by datasets used

This section summarizes the selected studies according to the datasets considered. As shown in Tables C.9–C.14 (see Appendix C), the intrusion datasets vary on each study. It is worth mentioning that it would be necessary to use multiple datasets to construct a detection model. Therefore, it would prove the generalizability of the proposed model in different environment settings. Table 2 shows the number of IDS datasets that have been used in the selected studies. Note that a particular study uses only a single dataset, while the other studies have utilized multiple datasets. A large body of work has been performed using NSL-KDD, KDD Cup 99, and UNSW-NB15 datasets; however, there are a few attempts available in the selected studies on applying ensemble learners for Honeypot, CAIDA, and AWID datasets. What is more, a number of ensemble techniques have been applied on non-public and specific datasets such as LLS-DDoS, Indoor WSN, IoT Botnet, etc.

5.4. Mapping selected studies by ensemble methods

In this section, we categorize the existing studies that use at least one ensemble method for IDSs. As mentioned earlier, Zhou [11] suggests that ensemble methods can be grouped into two main families, i.e. homogeneous and heterogeneous. Table 3 summarizes and classifies the selected studies in terms of such two categories.

In line with our mapping studies, the vast majority of ensemble-based IDSs approaches were built using homogeneous learners. Random forest (RF) [150] is the most common algorithm, followed by bagging [13] and Adaboost [151]. It is worth mentioning that several tree ensemble techniques have also been taken into account in IDSs domain. These include GBM [152] and XGBoost [153]. Besides, rotation forest [154] was another ensemble learning which has shown significant contribution in the purview of IDSs. Furthermore, there has been a great interest on the utilization of heterogeneous ensembles such as majority voting (e.g., dictatorship) [155] and stacked generalization (e.g., stacking) [156]. The following part is devoted to briefly discuss each selected study with respect to ensemble configuration, base learner used, and so on.

Verma and Ranga [146] compared several ensemble learners, i.e. bagging, Adaboost, random subspace, and RUSBoost for network IDSs, where decision tree and a discriminant learner were used as base learners. Tama et al. [85] proposed a two-stage ensemble learning that was made up of two ensemble learners, i.e. rotation forest and bagging for anomaly-based IDSs. A conjunctive rule classifier was employed as a base learner. Subudhi and Panigrahi [138] compared bagging, boosting, and stacking for IDSs. Bagging and boosting were used to improved the performance of decision tree, while several individual learners, i.e. naive Bayes, k-nearest neighbors (k-NN), rule induction, and decision tree were combined to build a stack ensemble. Similarly, Illy et al. [86] conducted a comparative analysis of several ensemble learning algorithms such as bagging, boosting, RF, and voting both for anomaly and misuse-based IDSs. Bagging of decision tree was the best performer for binary classification, whilst a voting ensemble using k-NN, RF, bagging, and boosting of decision trees was the best-performing ensemble for multi-class classification.

Al-Mandhari et al. [47] investigated the use of different machine learning (ML) algorithms in order to overcome the misclassification problems of KDD Cup 99 dataset. According to their results, RF was the best-performing algorithms. Zwane et al. [121] analyzed seven ML algorithms, i.e. multilayer perceptron (MLP), Bayesian network (BN), support vector machine (SVM), Adaboost, RF, bagging, and decision tree (J48) for addressing IDSs issues in tactical wireless networks. The results indicated that ensemble learning outperformed single classifiers w.r.t accuracy, AUC, TPR, and FPR metrics. Vinutha and Poornima [78] discussed several ensemble techniques, i.e. Adaboost, bagging, and stacking for IDSs, in which J48 was used as a base learner. According to the results, Adaboost improved the classification accuracy on benchmark dataset, e.g. NSL-KDD. Vaca and Niyaz [118] applied several ensemble learning methods on Wi-Fi intrusion dataset, called AWID. The results indicated that RF was better than bagging, ExtraTrees, and XGBoost when it was used to identify whether a record was an attack or normal.

Pham et al. [79] argued that bagging of J48 produced the best performance in terms of classification accuracy and FPR metric when working with the subset of 35 selected features. Kaur and Hahn [136] explored the use of a bagging ensemble for IDSs in smart grid. The result was quite similar to the one obtained by Pham et al. [79], where bagging of J48 gave a better performance w.r.t a recall metric. Ghafir et al. [137] proposed a ML-based approach to detect advanced persistent threat (APT), which is the most serious type of cyberattack. Experimental results show that the best classification algorithm was the linear SVM, outperformed ensemble classifiers, i.e. boosting and bagging trees. Gautam and Doegar [44] used Adaboost for anomaly-based IDSs, where the results concluded that the ensemble approach was better than any classifiers as an individual learner. Dahiya and Srivastava [122] utilized two different feature selection techniques, i.e., canonical correlation analysis and linear discriminant analysis, and several ML algorithms were applied on UNSW-NB15 intrusion dataset. The study concluded that random tree classifier was the winner under various performance metrics.

Table 3
Summary of selected studies utilizing ensemble methods for IDSs.

Ensemble family	Ensemble scheme	Selected studies	#
Homogeneous	Bagging	[30,40,44,47,54–60,70,78–80,85,86,94,97,103,117,118,120–122,125,134,136–138,141,142,146]	33
	Boosting	AdaBoost [27,31,32,40–42,47,48,50–52,54,55,58,60,70,71,78–80,86,87,96,97,104,111,120,121,125,126,129,138,146,148,149,157]	35
		RUSBoost [120,137,146]	3
		Not mentioned [137]	1
		LogitBoost [94,103,120]	3
		GentleBoost [53,120]	2
		LPBoost [62]	1
		RealBoost [53,56]	2
		MultiBoost [56]	1
		CatBoost [107,147]	2
		ModestBoost [53]	1
	Random subspace	[46,137,146]	3
	Rotation forest	[55,56,70,85,110,111]	6
	Tree ensemble	Random forest [28,29,32–36,40–42,45,47,49,52,56,60,61,63–66,70,72,73,80,86,93,94,97,99,103,108,112,114–119,121–123,125,127,131,132,134,135,140,146–148]	52
		Gradient boosting tree [35,36,52,74,81,82,88,93,95,97,99,104,107,118,119,126,130,147]	18
		GAR-F [66]	1
		Weighted-random forest [131]	1
		ETC [104]	1
	Dagging	[141]	1
Heterogeneous	Stacking	[40,43,47,60,64,67,78,96,100,102,124,127,128,138,139,143,147]	17
	Voting	Average probability voting [37,55,56,68,89,105,106,108]	8
		Weighted majority voting [38,39,69,75,83,90]	6
		Not mentioned [28,125,127,144,157]	5
		Weighted sum voting [70,133]	2
		Majority voting [43,55,56,68,70,77,84,86,91,92,101,106,109,113,117,135,140,143]	18
		Maximum probability voting [68,106,135]	3
		Product probability voting [68,135]	2
		Sum probability voting [76]	1
		Minimum probability voting [68,145]	2
		Median probability voting [106,145]	2
		Bayesian [98]	1

Al-Jarrah et al. [80] proposed a semi-supervised multi-layered clustering (SMLC) model for IDSs. The performance of SMLC was compared with that of supervised ensemble ML models and a well-known semi-supervised model (i.e., tri-training). Timčenko and Gajin [120] evaluated several ensemble classifiers and compared their learning capabilities on UNSW-NB15 dataset. The obtained results had indicated that bagged tree and GentleBoost performed with highest accuracy and AUC values, while RUSBoost had the lowest performance. Miller and Busby-Earle [70] explored an approach to classify intrusion, called multi-perspective machine learning (MPML). The proposed method utilized naive Bayes algorithm to combine the results of base classifiers in ensemble. Kushwaha et al. [40] determined the most appropriate feature selection algorithm to select the relevant features of KDD Cup 99 dataset. Thirty features were successfully chosen, while various classifiers were also used for classification. Ajaeiya et al. [134] introduced a lightweight flow-based IDSs in a SDN. The study was concluded that RF classifier was able to detect multiple types of attacks and separate those attacks from the network traffic.

Ponomarev and Atkison [141] proposed an approach to detect the intrusion into network attached industrial control systems (ICSs) by measuring and verifying network telemetry. The results indicated that the bagging of REPTree classifier was able to reach the highest accuracy when classifying traffic between two computers of different hardware configuration. Mehetrey et al. [30] investigated a collaborative IDSs based on an ensemble method. In the proposed system, tasks are distributed among virtual machines, individual results are then blended for final adaptation of the learning model. Medina-Pérez et al. [142] introduced a masquerader detection method, namely Bagging-TPminer, a one-class ensemble learning. The proposed method improved classification accuracy when compared to other classifiers. Alotaibi and Elleithy [117] built a misuse wireless local area network IDS. The proposed method used a majority voting to vote the class predictions of extra trees, RF, and bagging. Tama and Rhee [55] analyzed the performance of several ensemble learning schemes to detect DoS attack. The results indicated that an ensemble with average probability voting was the best-performing method w.r.t. accuracy metric.

In [56], a network anomaly detection was presented. The proposed method was built using PSO for attribute selection and the ensemble of C4.5, RF, and CART. Sreenath and Udhayan [57] aimed to provide an IDS technique using a bagging ensemble selection, which gave an excellent predictive performance for practical settings. Robinson and Thomas [125] demonstrated the effectiveness of ensemble learning for classifying DDoS attack. The detection model was developed by combining Adaboost and RF algorithms. Gaikwad and Thool [58] argued that bagging ensemble with REPTree as a base classifier exhibited the highest classification accuracy as well as the low false positive rate when it was applied on NSL-KDD dataset. Similarly, by using ensemble learning, Gaikwad and Thool [59] implemented an IDS technique based upon bagging of a partial decision tree classifier. Besides,

the dimension of input features were chosen using a genetic algorithm. Choudhury and Bhowal [60] compared several ML algorithms for categorizing network traffic. According to the experimental results, it could be concluded that RF and BN were suitable for such purpose.

Mazini et al. [87] proposed a new reliable hybrid method for an anomaly-based IDS using artificial bee colony and Adaboost algorithm. Results of the simulation on NSL-KDD and ISCXIDS2012 datasets confirm that this reliable hybrid method had a significant difference from other IDS, which are accomplished according to the same datasets. Jan [48] used a semi-parametric model for IDS, where the vector quantization technique were applied o the training data. The proposed mode is further improved by Adaboost algorithm. Bansal and Kaur [126] benchmarked various ML algorithms for detecting different types of DoS attack. Concerning the performance accuracy, XGBoost performed efficiently and robust in classifying the intrusion. Aljawarneh et al. [157] utilized a hybrid algorithm consist of several classifiers, i.e., J48, meta pagging, random tree, REPTree, Adaboost, decision stump, and naive Bayes for IDS. The experimental results revealed that the hybrid approach had a significant effect on the minimization of the computational and time complexity.

Vinayakumar et al. [41] evaluated the effectiveness of various shallow and deep networks on KDD Cup 99 and NSL-KDD dataset in both binary and multi-class classification settings. Tama and Rhee [111] proposed new approach of anomaly-based IDS in wireless network using multi-level classifier ensembles. The proposed method was based on the combination of two different ensemble learners, i.e. rotation forest and Adaboost. Mkuzangwe and Nelwamondo [71] used Adaboost-based IDS that uses decision stump as a weak learner to classify Neptune and normal connections. He et al. [42] proposed a SDN-enabled traffic anomaly detection method using two refined algorithms. Adaboost and RF classifier were used in the comparison for being compared with the proposed model. Yuan et al. [31] proposed a novel network anomaly detection method using a combination of a tri-training approach with Adaboost algorithm. The bootstrap samples of tri-training were replaced by three different Adaboost algorithms to create the diversity. Ni et al. [32] proposed a two-stage approach that consists of an unsupervised feature selection and density peak clustering to handle label lacking problems. In most case, RF and Adaboost could achieve better detection accuracy compared to other models, i.e. DT and SVM.

Sornsuwit and Jaiyen [27] took into account ensemble learning, e.g., Adaboost to improve the detection of U2R and R2L attacks. In addition, the correlation-based technique was used to reduce redundant features. Thaseen and Kumar [62] built a hybrid model for intrusion detection combining consistency feature selection and ensemble of weak classifiers, i.e., SVM and LPBoost. The proposed model outperformed individual classifiers, i.e., SVM and neural network. Tama and Rhee [110] examined different base algorithms when applying rotation forest for IDSs. Twenty different algorithms were included, in which the detection performances were assessed in terms of AUC metric. Li et al. [49] proposed an AI-based two-stage intrusion detection empowered by software-defined technology. Bat algorithm was firstly used to select the features, then RF with weighted voting mechanism was exploited to classify flows. Abdulhammed et al. [127] utilized various techniques for tackling imbalanced dataset to develop an effective IDS from CIDDS-001 dataset. The effectiveness of sampling methods on CIDDS-001 was carefully studied and experimentally evaluated through deep neural networks (DNNs), RF, voting, variational autoencoder, and stacking. Vigneswaran et al. [45] evaluated DNNs to predict attacks on IDS. For the sake of comparison, the training was performed on the same dataset with several machine learning algorithms, including ensemble learners, i.e., Adaboost and RF. Soheily-Khah et al. [114] proposed a hybrid IDS, namely kM-RF, which is a combination of K-means clustering and RF algorithm. A benchmark dataset (e.g., ISCH) was employed to evaluate the proposed model. Besides, a deep analysis was also conducted to study the impact of the importance of each feature defined in the pre-processing step.

Injadat et al. [115] utilized a Bayesian optimization method in order to improve the performance of anomaly-based detection systems using SVM-RBF, RF, and k-NN. In particular, the Bayesian optimization method is used to set the parameters of the classifiers through finding the global minimum of the corresponding objective function. Belouch et al. [123] used the UNSW-NB15 dataset (i.e., a recent public dataset for network IDSs) and identified that RF classifier provides better performance to classify whether the incoming network traffic was normal or an attack than the performance of SVM, naïve Bayes, and decision tree. Ahmad et al. [116] identified that extreme learning machine (ELM) can be a more appropriate ensemble technique for IDSs that are designed to analyze large-scale data (e.g., NSL-knowledge discovery and data mining (KDD) dataset) than RF and SVM. On the other hand, Primartha and Tama [72] validated that RF for IDSs substantially outperforms the similar ensemble model (i.e., ensemble of random tree + naïve Bayes tree) and other single classifiers (i.e., naïve Bayes and neural network) based on the experimental results using three public intrusion datasets (i.e., NSL-KDD, UNSW-NB15, and GPRS).

Kumar et al. [135] evaluated the performance of the ensemble machine learning methods based on RF, C4.5, ripple-down rule learner, repeated incremental pruning to produce error reduction (RIPPER), and partial decision tree (PART), which are developed to detect known and unknown mobile treats, and identified that the ensemble machine learning methods outperform individual classifiers. Jabbar et al. [108] proposed a novel ensemble classifier, which are based on RF and average one-dependence estimator (AODE), to classify network traffic as normal or malicious and validated that the proposed ensemble classifier provides high classification performance using Kyoto dataset. Shen et al. [46] argued that random subspace along with ELM as a base classifier can be used to improve the accuracy and robustness of an IDS. In addition, a bat algorithm was utilized to optimize the ensemble model. Belavagi and Muniyal [73] identified that RF outperforms SVM, Gaussian naïve Bayes, and logistic regression for classification of a normal behavior and four classes of attacks (i.e., denial of service (DoS), remote to local (R2L), probe and user to root (U2R) attacks) for intrusion detection. Rodda and Erothi [63] discussed a class imbalance problem in IDSs and also identified that RF outperforms naïve Bayes, Bayes network, and C4.5 for classification of a normal behavior and four classes of attacks using the NSL-KDD dataset.

Rathore et al. [33] proposed a Hadoop-based real-time IDS, which consists of four-layered IDS architecture (i.e., capturing layer, filtration and load balancing layer, processing or Hadoop layer, and decision-making layer), for high-speed big data environment. Major machine learning techniques (i.e., naïve Bayes, SVM, conjuctive rule, RF, REPTree, and C4.5) and DARPA, KDDCup99, and NSL-KDD datasets were used to validate that the proposed Hadoop-based real-time IDS outperforms the existing IDSs. Mishra et al. [119] proposed a robust security architecture (i.e., NvCloudIDS) that analyzes network traffic and process to detect intrusions at network and virtualization layer in cloud environment. In particular, they presented a novel ensemble machine learning algorithm that improves the performance of RF and reduces its overfitting problem. Milliken et al. [64] used a multi-objective genetic algorithm in order to determine Pareto-optimal ensembles of base-level classifiers for intrusion detection and validated that Pareto-optimal ensembles outperform the majority of base-level classifiers using the NSL-KDD dataset.

Masarat et al. [34] proposed a novel parallel RF algorithm in order to overcome shortcomings of an original RF algorithm in selecting features, selecting the optimal number of classifiers, and selecting the optimal number of random features for training and improve the performance of IDSs. Mabu et al. [65] provided a random-forests-based classifier using genetic network programming and applied it to IDSs. The experimental results based on the NSL-KDD dataset indicated that the proposed classifier outperforms SVM, C4.5, and RF. Kulariya et al. [35] identified that RF provides better performance (i.e., accuracy, sensitivity, and specificity) and requires less prediction time than logistic regression, SVM, naïve Bayes, and GB tree for detecting the attack traffic. While naïve Bayes requires less training time than RF, but the difference is not significant.

Kanakarajan and Muniasamy [66] developed a novel tree ensemble technique (i.e., GAR-F) through applying greedy randomized adaptive search procedure with annealed randomness and feature selection in order to enhance classification accuracy of IDSs. Junejo and Goh [132] investigated a behavior-based machine learning approach that models the physical process of the cyber–physical system (CPS) to detect any anomalous behavior or attack, which can change the behavior of the CPS. They validated the effectiveness of the proposed approach using nine machine learning techniques (i.e., neural network, SVM, LR, RF, C4.5, BFTree, Bayesian network, naïve Bayes, and k-NN) on Secure Water and Treatment (SWaT) testbed (i.e., a complete replicate of the physical and control components of a real modern water treatment facility). Gupta and Kulariya [36] proposed a framework for efficient and fast cyber security network intrusion detection based on Apache Spark. A two well-known feature selection algorithm (i.e., correlation-based feature selection and Chi-squared feature selection) and five classification techniques (i.e.,g logistic regression, SVM, RF, gradient boosted decision trees, and naïve Bayes) are used for developing the proposed framework.

Stevanovic and Pedersen [112] presented three traffic classification methods based on a capable RF classifier for detecting botnets, which represent one of the most serious threats to the Internet security recently. Ronao and Cho [131] proposed a RF-based method with weighted voting (i.e., weighted RF) for IDSs and identified that the proposed method with weighted voting provides more consistent performance than that with balanced voting. Liu et al. [140] proposed the intrusion detection framework named "Opprentice" that applies RF to acquiring realistic anomaly definitions as well as automatically combining and tuning various detectors.

Hedar et al. [61] proposed a new hybrid IDS based on accelerated genetic algorithm and rough set theory for data feature reduction as well as genetic programming with local search for data classification. In particular, they identified that data feature reduction contributes to improve classification performance and reduce memory and CPU time. Elekar [28] discussed combinations of machine learning techniques in order to improve an attack detection rate and reduce a false attack detection rate for IDSs. C4.5 with random tree, C4.5 with random forest, and random forest with random tree are considered as possible combination candidates and it is identified that the performance of C4.5 with random tree is better than others for both improving an attack detection rate and reducing a false attack detection rate. Zhou et al. [81] proposed a novel deep learning model for real-time cyber attack detection in the IoT environment. Especially, the proposed model uses large-scale data to generate high-level features and applies the pretrained model to boost the detecting speed of traditional machine learning techniques.

Zhang et al. [82] presented XGBoost based on a stacked sparse autoencoder network, which is used for learning the deep features of intrusion detection data in an unsupervised manner, for a network IDS. Yousefi-Azar et al. [74] proposed an unsupervised feature learning model for malware classification and network-based anomaly detection based on auto-encoder. Tama and Rhee [88] used a gradient boosted machine in order to improve detection performance of anomaly-based IDS. In particular, the optimal performance of gradient boosted machine is obtained through a grid search of training parameters.

Branitskiy and Kotenko [143] exploited the ensemble of adaptive binary classifiers for network anomaly detection. A decisive classification rule consists of majority voting, stacking, and combining the classifiers using the arbiter based on the dynamic competence regions. Branitskiy and Kotenko [43] developed and evaluated a hybrid approach in order to detect network attacks using a multi-level integration of traditional mechanisms and computational intelligence detection models, including neural networks, immune systems, neuro-fuzzy classifiers, and SVM. A simple voting technique, an improved stacking algorithm, a technique using the arbiter based on the dynamic areas of competence, and a classification tree with neural networks as nodes were considered for discussing hybridization schemes. Chand et al. [67] conducted a comparative analysis of the performance of SVM when it is stacked with other classifiers, such as Bayesian network, AdaBoost, logistic regression, k-NN, C4.5, RF, JRip, OneR, and SimpleCart for IDS. The experimental results using NSL-KDD indicated that stacking of SVM and RF provides better performance than others.

Salo et al. [89] proposed a novel hybrid dimensionality reduction method for intrusion detection through combining information gain and PCA as well as developing an ensemble classifier based on SVM, k-NN, and multilayer perceptron. Malik et al. [29] utilized a binary PSO to find more appropriate set of attributes, while RF was used as a classifier for network intrusion detection. Ying et al. [37] presented an ensemble learning model based on Bayesian network and random tree for IDS and identified that the proposed ensemble learning model outperforms base classifiers (i.e., Bayesian network and random tree) through the experiment using the KDDcup99 dataset. Gaikwad and Thool [68] proposed a novel architecture that consists of unstable base classifiers and combines the advantages of rule learners and decision trees with a voting rule combination method for IDSs.

Parhizkar and Abadi [106] presented a web-based anomaly detection approach based on an ensemble of one-class SVM classifiers with a binary artificial bee colony algorithm, which prunes the initial ensemble of one-class SVM classifiers and finds a near-optimal sub-ensemble. Moustafa et al. [90] proposed an intrusion detection technique for mitigating malicious events, such as botnet attacks against a domain name system, hyper text transfer protocol, and message queue telemetry transport protocols that are utilized in IoT networks. In particular, an AdaBoost ensemble learning method is used based on decision tree, naïve Bayes, and artificial neural network. Thaseen et al. [83] developed an intrusion detection model based on Chi-square feature selection and the ensemble of classifiers (i.e., SVM, modified naïve Bayes, and LPBoost). Majority voting ensemble for IDSs was discussed in [91,92,113]. Several weak learners as well as strong learners were considered as base classifiers of their proposed model. Sornsuwit and Jaiyen [50] compared AdaBoost algorithm with different base learner settings. The performance of proposed model was validated on multiple datasets.

Ludwig [75] designed the IDS using a neural network ensemble method, which is based on autoencoder, deep belief neural network, deep neural network, and an extreme learning machine, in order to classify the different network attacks. Lueckenga et al. [69] proposed a weighted vote classification method and a general weight calculation function for improving the detection performance of anomaly-based smart grid IDSs. Aburomman and Reaz [38] implemented an ensemble of LDA and PCA

for developing an efficient IDS. Aburomman and Reaz [39] also proposed a novel ensemble construction method using PSO and weighted majority voting in order to improve intrusion detection performance for IDSs.

Jabbar et al. [144] presented a cluster-based ensemble classifier using ADTree and k-NN for IDSs. Maglaras et al. [133] considered to combine social network analysis metrics and ensemble machine learning techniques for improving the performance for intrusion detection. Zaman and Lung [109] presented six machine learning techniques (i.e., k-means, k-NN, fuzzy c-means, naïve Bayes, SVM, and radial basis function) and an ensemble method for network traffic anomaly detection. Jabbar et al. [84] used naïve Bayes and ADTree for developing an novel ensemble classifier that is used for network IDSs. Kevric et al. [76] proposed an effective combining classifier method using tree-based algorithms, such as random tree, C4.5, and NBTree, for network intrusion detection. Bosman et al. [145] exploited a decentralized unsupervised online learning scheme incorporating local neighborhood information and an ensemble method for spatial anomaly detection in sensor networks. Tama and Rhee [77] considered an anomaly-based intrusion detection system by hybridizing three different feature selection techniques and an ensemble learning.

Abirami et al. [124] combined RF, SVM, and NB using a stacking algorithm, where LR was used as a meta-classifier for IDS. Bedi et al. [93] proposed an algorithm-level approach called I-SiamIDS, which is a two-layer ensemble for handling class imbalance problems. Cheng et al. [128] utilized a semi-supervised hierarchical stacking model for anomaly detection in IoT communication. Dash et al. [129] proposed a multiclass adaptive boosting ensemble learning-based model with the synthetic minority oversampling technique for anomaly detection in IoT network. Du and Zhang [51] applied a two-level selective ensemble learning algorithm for handling imbalanced datasets. Gormez et al. [147] compared and tuned several machine learning methods including ensemble models and autoencoder-based deep learning classifiers using Bayesian optimization. The methods were trained and tested both for binary and multi-class classification on Digiturk and Labris datasets.

Gupta and Agrawal [148] incorporated IDS model based on parallel ensemble using bagging for anomaly and misuse detection in computer networks. Hariharan et al. [94] analyzed the performance of an IDS model using individual and ensemble-based classification algorithms. The lowest error was given by RF classifier. Huan et al. [149] argued that the fusion of clustering undersampling and AdaBoost algorithm improved the effect of network traffic anomaly detection. Jafarian et al. [139] used a novel combined approach for anomaly detection. The method used NetFlow protocol for acquiring information and generating datasets, information gain ratio for selecting the relevant attributes, and stacking ensemble for detecting anomaly in SDN networks. A work in [95] proposed the PSO-XGBoost model for network IDS. A classification model based on XGBoost was developed and PSO was employed to adaptively search for the optimal structure of XGBoost.

Karatas et al. [52] used SMOTE to reduce the imbalance ratio of intrusion datasets. Synthetic samples were introduced for minor classes and the their numbers are increased to the average data size via the proposed technique. Kaur [96] compared and studied two ensemble methods, namely weighted voting based AdaBoost ensemble and stacking-based ensemble. An accuracy of 89.75% was achieved by stacking based ensemble for UNSW-NB15 test set. Li et al. [107] discussed an anomaly detection over HTTP traffic. The traffic data was processed using Word2vec algorithm to deal with the semantic gap and implement TF-IDF to construct a low-dimensional vector representation. Two boosting algorithms, i.e., CatBoost and LightGBM were also used for constructing the detection model. Liu et al. [97] examined specific attacks in the NSL-KDD dataset that can impact sensor nodes and networks in IoT settings. XGBoost algorithm ranks the first, outperforming the rest of the studied supervised algorithms.

Otoum et al. [98] employed a Bayesian combination technique to construct an ensemble model, where RF, DBSCAN, and RBM were used as base classifiers. Rai [99] implemented an ensemble learning by combining RF, GBM, and XGBoost for IDS. The proposed model outperformed DNN after applying a feature selection technique using GA. Similarly, Rajadurai and Gandhi [100] proposed a stacked ensemble system for classifying attacks, where RF and GBM were used as base classifiers. Shahraki et al. [53] evaluated and compared Real AB, Gentle AB, and Modest AB for network IDSs. The results showed that Modest AB has a higher error rate compared to Gentle AB and Real AB. Singh and Singh [130] proposed a novel ensemble hyper-tuned model that automatically and effectively detects IoT sensors attacks and anomalies. The proposed model was built on the basis of feature selection and ensemble technique.

Swami et al. [101] analyzed three voting-based ensemble models for detecting DDoS attacks against SDN. Tama et al. [102] proposed a stacked ensemble for anomaly-based IDS in a Web application. Validation result on the network IDS also ameliorate the ones achieved by several recent approaches. A study performed by Uzun and Balli [103] showed that RF classifier achieved the highest classification accuracy on NSL-KDD dataset. Verma and Ranga [104] used an ensemble method for securing IoT against DoS attacks. In addition, Raspberry Pi was used to evaluate the response time of classifiers on IoT specific hardware. Wei et al. [54] established an intrusion detection algorithm based on ensemble support vector machine with bag representation. Lastly, Zhou et al. [105] proposed several steps for their intrusion detection framework. In the first step, a CFS-BA algorithm was employed for feature reduction. Then an ensemble approach that combines DT, RF, and Forest PA was introduced for attack recognition.

5.5. Mapping selected studies by intrusion detection techniques

In this section, the selected studies are categorized into three different intrusion detection models, i.e. anomaly, misuse, and hybrid-based techniques by following the IDS taxonomy discussed in [10]. Fig. 6 denotes the distribution of the selected studies with respect to such categories. The vast majority of IDS technique is anomaly-based detection that accounts for 65.32%, while misuse and hybrid-based detection share 25.81% and 8.87% of the total studies, respectively. Tables C.9–C.14 summarize (in chronological order) 124 selected studies that involved at least one ensemble learner in their experiment. Besides, those tables provide for each study the information regarding ensemble scheme, base learners, feature selection used, validation techniques and datasets considered, and best results obtained. The tables are presented in Appendix C for maintaining the readability of the manuscript.

6. Empirical evaluation

We carried out an empirical benchmark to identify the relative performance of ensemble methods in IDSs to answer RQ_4. This section presents the rationale of conducting the benchmark, the classification algorithms, the intrusion datasets, the evaluation setup and performance measures, and provide the results of the comparison.

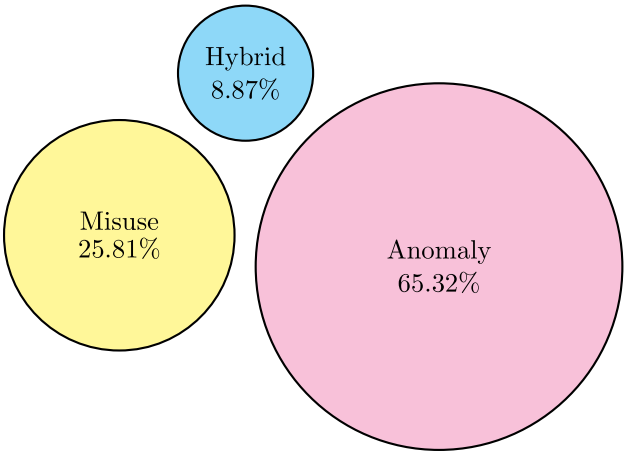

Fig. 6. Distribution of selected studies w.r.t three different IDS categories.

6.1. Rationale

Summarizing the above mapping study, ensemble learning-based IDSs have utilized a variety of schemes. The most popular choice has been a tree-based ensemble, e.g., random forest, which has advantages in terms of a reliable feature importance estimate, but less interpretable than a single decision tree. Other popular ensemble methods have been bagging, boosting, majority voting, stacking, and gradient boosting tree. An open research challenge in IDSs is the one of finding a suitable and high performing algorithm. This is a complex attempt as every intrusion dataset is likely to be very diverse, ranging from different scenario and network architecture. Given the variability of the dataset characteristics, it is essential to develop a comparative study of several ensemble methods, in order to support researchers and practitioners in selecting the best algorithm at hand.

In this section, we focus particularly on the issue of comparing an ensemble scheme, e.g., stacking [156], and the issue of designing the size of ensemble when homogeneous classifiers are used as base classifiers. Despite the fact that various ensemble schemes were utilized effectively by prior works, the choice of available schemes and base classifiers might need basic knowledge about the dataset. Besides, researchers are usually familiar with specific or well-known ensemble methods, therefore, ensemble methods are taken into account randomly without considering other less-common ensemble methods beyond the researchers' knowledge. Consistent with the 'no free lunch theorem', while some ensemble methods may perform better in a given intrusion dataset, the best-performing ensembles will vary among different datasets [158]. Hence, a benchmark of ensemble methods and base classifiers in IDSs is still currently lacking.

In order to distinguish this comparative study with another similar work, e.g., [102,159], we take into account a stack of classifier ensembles (SoE) architecture [102,156,160,161], where base classifiers are trained in a parallel manner. Unlike existing ensemble methods that use weak classification algorithms, i.e., DT, CART, NN, and random tree; SoE combines a number of homogeneous ensembles, i.e., RF [150], XGBoost [153], and GBM [152] to classify network traffic. Section 6.2 provides a detailed discussion about the procedure of constructing SoE for anomaly-based intrusion detection.

6.2. Classification algorithms

This benchmark involves a comparison between SoE and the base classifiers that build the ensemble. Several experiments are carried out using three different SoE architectures, in which each ensemble architecture is made up of some homogeneous learners as level-0 classifiers (e.g., base classifiers) and a combiner (e.g., level-1 classifier). While we run the experiment, various base classifiers were designed, indicating the size of each SoE architecture, i.e. 2, 5, 10, and 20 classifiers. For instance, S-RF-2, S-XGB-5, and S-GBM-10 is a SoE architecture consisting of two RFs, five XGBoosts, and ten GBMs, respectively. Algorithm 1 shows the procedure of constructing SoE. It is worth mentioning that the best possible learning parameters for each base classifier are obtained using *random* search [162], which has obvious advantages in terms of a reasonable computational cost when the search space is huge [163]. We specify the same learning parameters for XGBoost and GBM as they have the same principle of gradient boosting. All possible parameter values were searched as specified by a search space provided in Tables 4 and 5. In addition, a generalized linear model (GLM) is chosen as a level-1 classifier as other classifiers are deemed to be appropriate [164,165]. We briefly discuss the classifiers forming the SoE as follows.

Algorithm 1: Stack of ensemble with cross-validation

Setup:
Dataset $\mathcal{D}$ with k instances and l features, which is denoted as input matrix $\mathcal{X}$ and response matrix $\mathcal{Y}$.

$$k\left\{\overbrace{\left[\begin{array}{c}\\ \mathcal{X}\\ \\ \end{array}\right]}^{l}\right\}\left[\begin{array}{c}\\ \mathcal{Y}\\ \\ \end{array}\right]$$

Define $\mathcal{L}$ base classifiers, along with their optimal hyperparameters.
Define the level-1 classifier, e.g. GLM.

Train stack of ensemble:
Train each of the $\mathcal{L}$ base classifier on the training set.
Perform *stratified* 10-fold cross-validation on each base classifier.
Collect the prediction results, $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{\mathcal{L}}$
Collect $\mathcal{K}$ prediction values from $\mathcal{L}$ base classifiers and construct a matrix $\mathcal{K} \times \mathcal{L}$, which is later called as matrix $\mathcal{Z}$.
Along with original response vector $\mathcal{Y}$, train level-1 classifier: $\mathcal{Y} = f(\mathcal{Z})$.

$$k\left\{\left[\begin{array}{c}\\ \mathcal{S}_1\\ \\ \end{array}\right]\ldots\left[\begin{array}{c}\\ \mathcal{S}_{\mathcal{L}}\\ \\ \end{array}\right]\left[\begin{array}{c}\\ \mathcal{Y}\\ \\ \end{array}\right]\rightarrow\mathcal{K}\left\{\overbrace{\left[\begin{array}{c}\\ \mathcal{Z}\\ \\ \end{array}\right]}^{\mathcal{L}}\right\}\left[\begin{array}{c}\\ \mathcal{Y}\\ \\ \end{array}\right]\right.$$

Prediction on new testing sets:
Get the prediction results from base classifiers and feed into level-1 classifiers.
Get the final stack of ensemble prediction, $\mathcal{O}_f$.

- Random forest (RF) [150]
  RF is an improved version of bagging [13], where a random selection method is utilized for tree construction. The concept of randomness lies into two different paradigms: (i) a random sampling of training instances when generating a tree and (ii) a random subset of features taken into consideration when splitting nodes. A lower bias and variance of the individual tree can be achieved by a non-pruning strategy, in which the trees are fully generated. The strategy of combining multiple trees has obviously benefits in terms of prediction accuracy improvement and over-fitting reduction.

Table 4
Hyperparameter settings for GBM and XGBoost.

Learning variable	Search space	Final hyperparameter value			
		UNSW 2018 IoT Botnet	CICIDS2017	NSL-KDD	UNSW-NB15
sample_rate	{0.20, 0.21, . . . , 1.00}	0.50	0.64	0.57	0.58
max_depth	{1, 2, . . . , 30}	14	7	8	12
col_sample_rate_per_tree	{0.20, 0.21, . . . , 1.00}	0.80	0.89	0.84	0.68
col_sample_rate_change_per_level	{0.90, 0.91, . . . , 1.10}	0.99	1.02	1.04	1.1
min_rows	$2^{\{0,1,\ldots,\log_2(nts)-1\}}$	4	2	64	1024
nbins	$2^{\{4,5,\ldots,10\}}$	1024	16	256	16
nbins_cats	$2^{\{4,5,\ldots,12\}}$	512	128	16	4096
min_split_improvement	$\{0, \ldots, 1 \times 10^{-4}\}$	1×10^{-06}	0	0	0
n_trees	–	500	500	500	500
histogram_type	{UA, QG, RR}	UA	UA	QG	RR

List of abbreviation. nts: number of instances in training set; UA: uniform adaptive; QG: quantiles global; RR: round robin.

Table 5
Hyperparameter settings for RF.

Learning variable	Search space	Final hyperparameter value			
		UNSW 2018 IoT Botnet	CICIDS2017	NSL-KDD	UNSW-NB15
sample_rate	{0.20, 0.21, . . . , 1.00}	0.71	0.48	0.48	0.95
max_depth	{1, 2, . . . , 30}	17	8	12	11
col_sample_rate_per_tree	{0.20, 0.21, . . . , 1.00}	0.73	0.84	0.84	0.73
col_sample_rate_change_per_level	{0.90, 0.91, . . . , 1.10}	0.96	1.02	1.02	1.07
min_rows	$2^{\{0,1,\ldots,\log_2(nts)-1\}}$	16	32	32	16
nbins	$2^{\{4,5,\ldots,10\}}$	32	32	32	512
nbins_cats	$2^{\{4,5,\ldots,12\}}$	16	32	32	256
min_split_improvement	$\{0, \ldots, 1 \times 10^{-4}\}$	0	0	0	1×10^{-06}
n_trees	–	500	500	500	500
histogram_type	{UA,QG,RR}	RR	QG	QG	QG

List of abbreviation. nts: number of instances in training set; UA: uniform adaptive; QG: quantiles global; RR: round robin.

In addition, since every individual tree is constructed on a small number of variables, RF has less computational power. The computational task can also be increased through a simpler implementation of parallel computing.

- Gradient boosting machine (GBM) [152]
  GBM uses weak classification models (e.g., classification and regression tree (CART) [166]) in an iterative way. Gradient is utilized to minimize a loss function. In each round of training, the weak learner is built and its predictions are compared to the actual case. The difference between prediction and the actual case denotes the error rate of the model. The error can be further used to calculate the loss function. Given a dataset $\mathcal{D}$ with m samples and s variables $\mathcal{D} = (x_i, y_i)(|\mathcal{D}| = m, x_i \in \mathcal{R}^s, y_i \in \mathcal{R})$, a tree ensemble employs $\mathcal{L}$ additive function to predict the final output [152].

$$\hat{y}_i = \phi(x_i) = \sum_{l=1}^{\mathcal{L}} f_l(x_i), f_l \in \mathcal{F} \tag{1}$$

  where the space of CART (classification and regression trees) is defined as: $\mathcal{F} = f(x) = w_{p(x)}(p : \mathcal{R}^s \rightarrow \mathcal{T}, w \in \mathcal{R}^{\mathcal{T}})$. The p represents the configuration of each tree that maps a sample to an appropriate leaf index. $\mathcal{T}$ represents the tree size, while f_k is a stand-alone tree configuration p and leaf weight w. The decision guidelines in the trees (p) is utilized to predict a given sample into the leaves and compute the outcome through the total score in the corresponding leaves (w).
- Extreme gradient boosting (XGBoost) [153]
  XGBoost is a specific implementation of the gradient boosting algorithm that can be used for classification and regression predictive modeling problems. It employs more accurate approximations to find the best models and an advanced regularization technique, which enhances model generalization and reduces model complexity. A faster training speed than other gradient boosting implementations is the additional advantage of XGBoost.
- Generalized linear model (GLM)
  GLM is used as a level-1 classifier, e.g. combiner. Depending on distribution and function, GLM deals with either regression or classification tasks [167]. Due to the fact that an anomaly-based IDS is a binary classification problem, logistic regression is used to classify network traffic either normal or anomaly. Let x be a sample and y be an outcome category, logistic regression models the probability of x to y, where the fitted model $\hat{y}$ can be calculated as:

$$\hat{y} = Pr(y = 1|x) = \frac{e^{x^T\beta+\beta_0}}{1 + e^{x^T\beta+\beta_0}} \tag{2}$$

Furthermore, since ensemble learners mentioned above are constructed using some weak learners, the following individual weak learners are also taken into consideration in the benchmark. This makes the benchmark fairer and more reasonable.

- Decision tree (J48) [168]
  J48 is an implementation of classical decision tree algorithm, e.g. C4.5 in an open source data mining tool, called Weka [169]. J48 builds classification rules in the form of a tree-like structure, which made up of a root and a number of nodes. Each node denotes a class label, while instances are registered to nodes according to the impurity level of the class label distribution. Default learning parameters were used in the experiment that are set as follows. Confidence factor: 0.25, tree-pruning strategy was used, number of folds used for reduced-error pruning: 3, and minimum number of instances per leaf: 2.
- Credal decision tree (C-DT) [170]
  C-DT works based on imprecise probabilities and uncertainty measures in splitting the attribute at each branching node when constructing trees. Learning parameters are set to default for each dataset. The parameter used in Dirichlet model: 1.0, maximum tree depth: -1, minimum weight of the instances in a leaf: 2.0, and amount of fold used for pruning: 3.

- Classification and regression tree (CART) [166]
  CART is trained in a recursive binary splitting manner to generate the tree. Binary splitting is a numerical process in which all the values are organized, and different split points are tested using a cost function. The split with the lowest cost is chosen. We specify the learning parameters as follows. Minimal number of observations at the terminal nodes: 2, number of folds in the internal cross-validation: 5, and minimal-cost complexity is used.
- Random tree (RT) [150]
  RT develops the tree by choosing K randomly attributes at each node without pruning. Some learning parameters are specified as follows. Number of randomly chosen attributes: 0, unclassified instances are not allowed, limited depth of the tree, minimum weight of the instances in a leaf: 1.0, and amount of data used for backfitting: 0.

6.3. Intrusion detection datasets

In this section, we briefly discuss the datasets that are commonly used in IDS. NSL-KDD and UNSW-NB15 are considered for network packets-based analysis, while CICIDS 2017 is used for Web traffic-based analysis. Lastly, UNSW 2018 IoT Botnet is a dataset that is specifically used for IoT network-based analysis.

- NSL-KDD [171]
  The NSL-KDD dataset is an improved version of its predecessor, i.e. KDD Cup 99. The dataset was built to overcome some critical issues in KDD Cup 99 such as redundant samples, unreliable, and bias results (e.g. the results are too optimistic) when applying machine learning algorithms. The dataset comprises 41 input features and one class label feature. In our benchmark analysis, a number of training records (i.e., 125,973 samples) were used for creating the classification model, where the numbers of records representing anomaly and normal classes are 67,343 and 58,630 samples, respectively. Concerning evaluation procedure, an independent testing set is also taking into consideration. The testing set consists of 22,544 samples.
- UNSW-NB15 [172]
  The UNSW-NB15 dataset was developed using IXIA PerfectStorm tool for generating real-world modern normal network packets along with synthetic attack activities. The dataset is made up of 43 features including a class label attribute, while 37,000 normal and 45,332 anomaly samples are used in the experiment, respectively. In addition, a specialized testing set (i.e., 175,341 instances) is utilized for evaluating the classifiers' performance.
- CICIDS 2017 [173]
  The CICIDS 2017 dataset was obtained by generating realistic benign background traffic using B-profile system. Some available protocols such as HTTP, HTTPS, FTP, SSH, and email were included in the dataset, providing a complete traffic dataset with attack diversity for benchmarking. The dataset consists of 78 input features with one class label feature. Besides, it is a highly imbalanced dataset, where the numbers of benign and malicious samples are 168,186 and 2180 samples, respectively. The original dataset is split into two parts, i.e. training and testing sets with a ratio of 80% and 20%, respectively.
- UNSW 2018 IoT Botnet [174]
  The UNSW 2018 IoT Botnet dataset was built by incorporating a combination of normal and botnet traffic. Simulated network traffic was acquired using Ostinato and Node-red tool. In this comparison study, we take into account 5% of the entire generated samples (i.e., 2,934,817 samples) with 16 input features. The numbers of normal and attack samples are 370 and 2,934,447 samples, respectively. Besides, a total of 733,705 samples is utilized as an independent testing set.

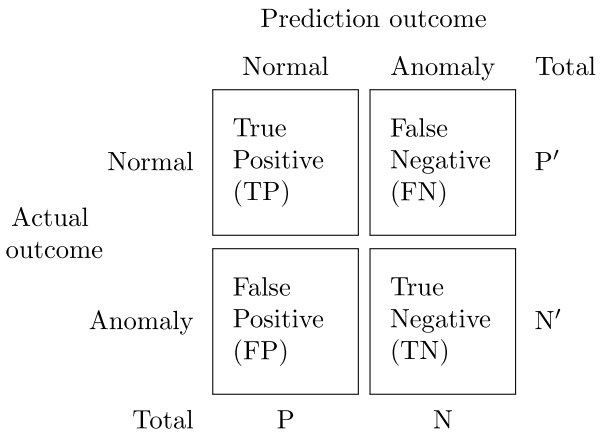

Fig. 7. A contingency table in case of solving a binary classification problem.

6.4. Evaluation setup

6.4.1. Performance metrics

Since we deal with some imbalanced datasets, we take into account Matthews correlation coefficient (MCC) [175]. To the best our knowledge, this study is the first of utilizing MCC metrics in evaluating IDS algorithms. MCC has been used to evaluate the performance of a binary classification problem [176] that is calculated as following.

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \in \{-1, 1\} \quad (3)$$

where a value of -1 denotes a completely wrong prediction and a value of $+1$ implies a perfectly correct prediction. $MCC = 0$ means that the classification model is no better than a random guessing, where there is no correlation between model predictions and the actual results. MCC takes into account all four values in a contingency table shown in Fig. 7. Moreover, three other metrics that are frequently used for anomaly-based IDSs are taken into consideration. These are accuracy (ACC), false positive rate (FPR), and area under ROC curve (AUC) that can obtained as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \in \{0, 1\} \quad (4)$$

$$FPR = \frac{FP}{FP + TN} \in \{0, 1\} \quad (5)$$

$$AUC = \int_0^1 \frac{TP}{TP + TN} d\frac{FP}{FP + FN} = \int_0^1 \frac{TP}{P} d\frac{FP}{N} \quad (6)$$

6.4.2. Statistical significance tests

In order to provide an unbiased benchmark among the classification algorithms, several significance tests must be taken into consideration. Such tests have become a standard in machine learning research, where multiple algorithms and datasets are typically involved [177–179]. The following statistical tests are used in this study.

- Friedman test [180]
  This is a non-parametric test, in which the differences between the classification algorithms μ_δ are tested in terms

Table 6
Results of average Friedman rank, an omnibus test, and Finner's method where GBM is a control algorithm.

	J48	C-DT	CART	RT	RF	GBM	XGB	S-RF-2	S-RF-5	S-RF-10	S-RF-20	S-GBM-2	S-GBM-5	S-GBM-10	S-GBM-20	S-XGB-2	S-XGB-5	S-XGB-10	S-XGB-20
Average rank	14.75	15.75	15.50	17.25	8.50	**4.25**	7.75	9.75	9.25	9.50	9.25	8.63	9.38	7.13	*7.38*	9.25	9.25	8.75	8.75
p-value												**0.089**							
Finner test	**S**	**S**	**S**	**S**	NS	N/A	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS	NS

S: significant; NS: not significant.

Table 7
Relative differences (%) of S-GBM-10 and GBM w.r.t MCC score. (For instance, S-GBM-10 performance is 2.03% lower than GBM on NSL-KDD dataset.)

Classifier I	Classifier II	NSL-KDD	UNSW-NB15	CICIDS-2017	UNSW-2018 IoT Botnet
S-GBM-10	GBM	−2.03	−0.06	0.00	−2.36
GBM	CART	33.11	6.64	0.09	2.42

of MCC metrics. The objective is to evaluate the two types of hypotheses, namely, null hypothesis ($\mathcal{H}_0$) and alternative hypothesis ($\mathcal{H}_\alpha$). The $\mathcal{H}_0$ means that there are no performance differences between the benchmarked classifiers ($\mu_\delta = 0$), while $\mathcal{H}_\alpha$ is otherwise ($\mu_\delta \neq 0$). The initial phase in calculating the test statistic is to convert the corresponding performance results to ranks. Therefore, it ranks the classifiers for each dataset independently, the best performing classifier is assigned with the rank of 1, the second best rank 2, etc. Let l, m, and $\mathcal{R}_i$ be the number of datasets, the number of classifiers, and average rank of classifiers, respectively. Under the null hypothesis (e.g., $\mathcal{R}_i$ should be equal), the Friedman statistic $\mathcal{X}_F^2$ is specified as:

$$\mathcal{X}_F^2 = \left[\frac{12}{lm(m+1)} \sum_{i=1}^{m} \mathcal{R}_i \right] - 3l(m+1) \tag{7}$$

- Finner post-hoc test [181]
  The test is a *p*-value adjustment in a step-down manner. Let $p_1, p_2, \ldots, p_{m-1}$ be the ordered *p*-values in increasing order, so that $p_1 \leq p_2 \leq \cdots \leq p_{m-1}$, and $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_{m-1}$ be the corresponding hypotheses. It rejects $\mathcal{H}_1$ to $\mathcal{H}_{m-1}$ if m is the smallest integer. Due to its simplicity and power, Finner test is a good choice in general.

To sum up, the procedures in conducting statistical significance tests discussed in this study can be elaborated as follows.

- Calculate the classifiers' rank for each dataset using Friedman rank with respect to their MCC scores in an increasing order, from the best performer to the worst performer.
- Calculate the average rank of each classifier over all datasets. The best-performing classifier is determined by the lowest value of Friedman rank. Note that the merit is inversely proportional to numeric value.
- Calculate *p*-value from an omnibus test, e.g., Friedman test.
- If the Friedman test demonstrates significant (*p*- value < 0.1 in our case), run the Finner's method. It is carried out based on a pairwise comparison, where the best-performing algorithm is used as a control algorithm for being compared with the remaining algorithms.

6.5. Result and discussion

In this section, we discuss the results of all experiments. We used a total of 19 classification algorithms, considering the fact that twelve different SoEs, three individual ensemble learners, and four weak learners were incorporated in experiment. Besides, four intrusion datasets were taken into account, thus leading to a total of 76 experiment combinations. Two machine learning tools, i.e. Weka [169] and H_2O [182] were utilized for running the classification task. In addition, all codes were implemented

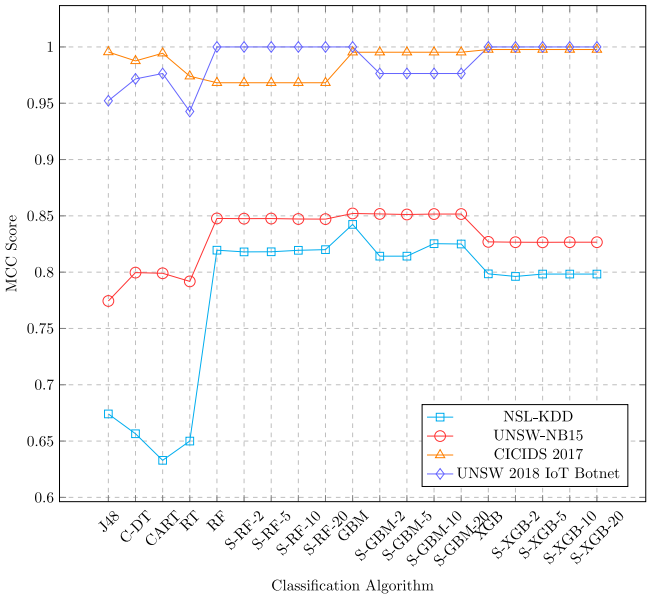

Fig. 8. Performance results of all classifiers for each intrusion dataset w.r.t. MCC metric.

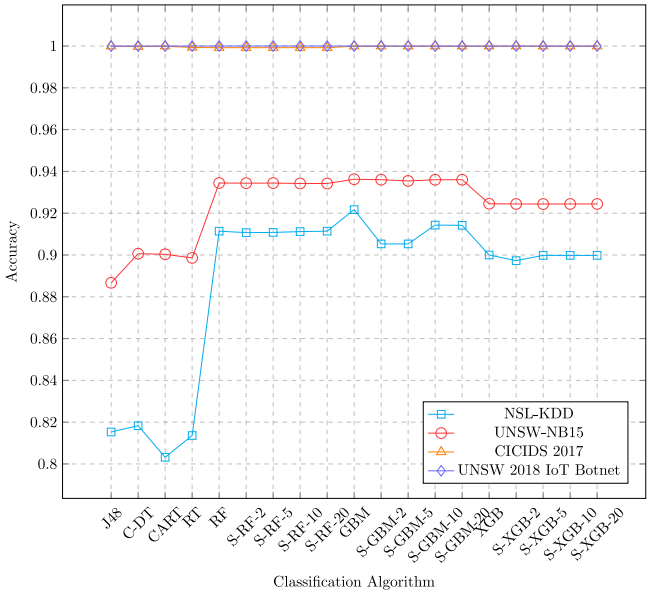

Fig. 9. Performance results of all classifiers for each intrusion dataset w.r.t. accuracy metric.

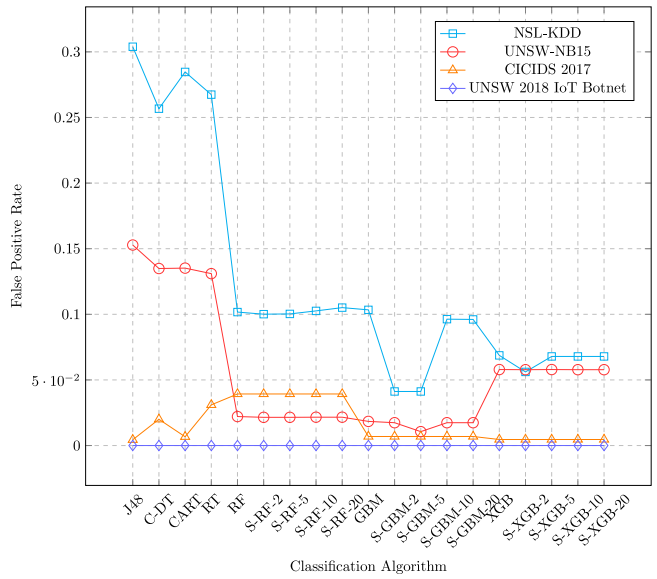

Fig. 10. Performance results of all classifiers for each intrusion dataset w.r.t. false positive rate metric.

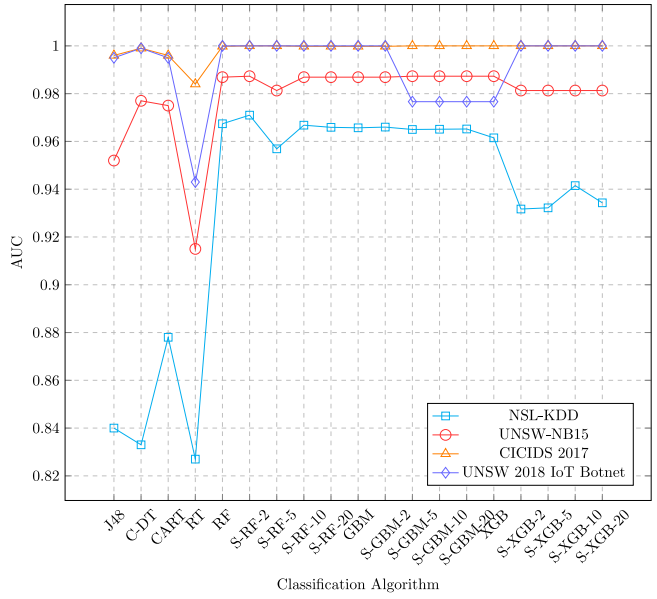

Fig. 11. Performance results of all classifiers for each intrusion dataset w.r.t. AUC metric.

in R and Python on a machine with Linux operating system, 32 GB memory, and Intel Xeon processor.

Figs. 8–11 show the average performance of all algorithms with respect to MCC, accuracy, FPR, and AUC metrics, respectively. The average performance results provided in these figures as well as the results of Friedman rank test (see Table 6) confirm that GBM is superior compared with other remaining classifiers, followed by S-GBM-10 and S-GBM-20. To further assess the performance differences among classifiers, an omnibus test using Friedman is undertaken. The test discovers significant ($p = 0.089 < 0.1$), meaning that there exists at least one classifier that performs differently. Therefore, since the test detects significance, Finner test is then applied. GBM is chosen as a control classifier for being confronted against the rest classifiers. Finner test reveals that GBM is significantly better than J48, C-DT, CART, and RT.

What is more, Table 7 contrasts the relative differences (%) of the best-2 performing algorithms. Surprisingly, S-GBM-10 could not obviously outperform GBM when applying on several datasets, i.e. NSL-KDD, UNSW-NB15, and UNSW-2018 IoT Botnet. In contrast, GBM could beat CART on the entire datasets. This is not surprising as GBM is built using multiple CARTs.

7. Threat to validity

Concerning internal validity, omitting important studies and researcher's bias during the process of paper inclusion might be inevitable. In this review, the search was restricted to well-known indexing database services. To reduce the risk associated with repealing important studies, the search keywords were derived from the keywords appear in some papers. Moreover, researcher's bias might affect the correctness of data extraction and mapping procedure. To solve this issues, a consensus mechanism among the authors was adopted to make the final selection and classification of the studies.

8. Conclusion

A systematic mapping study and comparative analysis of ensemble learning for intrusion detection systems were explored in this paper. The following part discusses the RQs and provides answers for them.

- RQ_1: What is the current trend in ensemble learning-based IDSs? This study revealed that there has been a great interest in applying random forest classifier for IDSs. This is because the implementation of random forest is diverse and almost effortless to apply for. For instance *Caret*, *Boruta*, *VSURF*, etc are the example of random forest implementation in *R*.
- RQ_2: What types of ensemble learning methods have been commonly used to cope with the issues arise in IDSs? The vast majority of ensemble learning discussed in this study is homogeneous ensemble, where random forest, bagging, and boosting were the most prevalent ensemble techniques. Furthermore, majority voting and stacking architecture were frequently utilized, particularly when heterogeneous classifiers were considered.
- RQ_3: What types of IDS techniques that are developed most? According to this mapping study, an anomaly-based intrusion detection system was the most common technique that accounts for two-thirds of the total publications.
- RQ_4: What is the relative performance of ensemble learning methods as compared to single classification algorithms? This study explores the relative performance of ensemble learning in comparison with individual classifier. It is revealed that ensemble learning has brought significant improvement over individual classifiers. However, it is not always the case, since it depends on various factors such as base classifiers, voting schemes, etc.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1F1A1059346). This work was supported by the 2020 Research Fund (1.180090.01) of UNIST (Ulsan National Institute of Science and Technology).

Table B.8
Distribution of selected studies w.r.t publication outlets.

No	Publication outlet	Type	#	%
1.	International Conference on Internet of Things: Smart Innovation and Usages	Conference	1	0.81%
2.	IEEE Access	Journal	6	4.84%
3.	Journal of King Saud University: Computer and Information Sciences	Journal	2	1.61%
4.	IEEE Wireless Communications and Networking Conference	Conference	1	0.81%
5.	Advances in Intelligent Systems and Computing	Book chapter	3	2.42%
6.	International Conference on Intelligent and Innovative Computing Applications	Conference	1	0.81%
7.	IEEE International Symposium on Network Computing and Applications	Symposium	1	0.81%
8.	Australasian Computer Science Week Multiconference	Conference	1	0.81%
9.	ACM Cybersecurity Symposium	Symposium	1	0.81%
10.	Future Generation Computer Systems	Journal	1	0.81%
11.	International Conference on Cloud Computing, Data Science & Engineering	Conference	1	0.81%
12.	Procedia Computer Science	Conference	5	4.03%
13.	Digital Communications and Networks	Journal	1	0.81%
14.	IEEE International Conference on Intelligent Computer Communication and Processing	Conference	1	0.81%
15.	International Conference on Machine Learning and Soft Computing	Conference	1	0.81%
16.	IEEE Region 10 Conference	Conference	1	0.81%
17.	IEEE Symposium on Computers and Communications	Symposium	1	0.81%
18.	IEEE Transactions on Dependable and Secure Computing	Journal	1	0.81%
19.	International Conference on Collaboration Technologies and Systems	Conference	1	0.81%
20.	Soft Computing	Journal	1	0.81%
21.	IEEE Long Island Systems, Applications and Technology Conference	Conference	1	0.81%
22.	International Workshop on Information Security Applications	Workshop	1	0.81%
23.	Lecture Notes in Electrical Engineering	Book chapter	2	1.61%
24.	IEEE International Conference on Engineering and Technology	Conference	1	0.81%
25.	IEEE Recent Advances in Intelligent Computational Systems	Conference	1	0.81%
26.	International Conference on Computing Communication Control and Automation	Conference	1	0.81%
27.	International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials	Conference	1	0.81%
28.	Communications in Computer and Information Science	Book chapter	2	1.61%
29.	Journal of Computational Science	Journal	2	1.61%
30.	International Conference on Advances in Computing, Communications and Informatics	Conference	2	1.61%
31.	International Conference on Systems and Informatics	Conference	1	0.81%
32.	IEEE Internet of Things Journal	Journal	4	3.23%
33.	ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems	Conference	1	0.81%
34.	International Conference on Applied Cryptography and Network Security	Conference	1	0.81%
35.	International Conference on Information Technology and Electrical Engineering	Conference	1	0.81%
36.	International Conference on Green Engineering and Technologies	Conference	1	0.81%
37.	Asia Joint Conference on Information Security	Conference	1	0.81%
38.	IEEE Sensors Letters	Journal	1	0.81%
39.	International Conference on Computing, Communication and Networking Technologies	Conference	1	0.81%
40.	International Conference on Data Intelligence and Security	Conference	1	0.81%
41.	IEEE Global Communications Conference	Conference	1	0.81%
42.	International Conference on Data and Software Engineering	Conference	1	0.81%
43.	International Conference for Internet Technology and Secured Transactions	Conference	1	0.81%
44.	International Conference on Electrical, Electronics, and Optimization Techniques	Conference	1	0.81%
45.	The Journal of Supercomputing	Journal	3	2.42%
46.	IEEE Symposium Series on Computational Intelligence	Symposium	1	0.81%
47.	Artificial Life and Robotics	Journal	1	0.81%
48.	International Conference on Communication and Signal Processing	Conference	1	0.81%
49.	International Conference on Frontiers in Intelligent Computing: Theory and Applications	Conference	1	0.81%
50.	ACM International Workshop on cyber–physical System Security	Workshop	1	0.81%
51.	International Conference on Cyber Situational Awareness, Data Analytics and Assessment	Conference	1	0.81%
52.	Lecture Notes in Computer Science	Book chapter	2	1.61%
53.	ACM Conference on Internet Measurement Conference	Conference	1	0.81%
54.	IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing	Conference	1	0.81%
55.	International Conference on Computer, Communication and Control	Conference	1	0.81%
56.	IEEE Conference on Computer Communications Workshop	Conference	1	0.81%
57.	IEEE International Conference on Communications Workshops	Conference	1	0.81%
58.	International Joint Conference on Neural Networks	Conference	2	1.61%
59.	Neural Computing and Applications	Journal	3	2.42%
60.	International Conference on Advances in Computing, Communication, & Automation	Conference	1	0.81%
61.	Computer Networks	Journal	2	1.61%
62.	IEEE International Conference on Software Engineering and Service Science	Conference	1	0.81%
63.	International Conference on Information and Communication Technology for Intelligent Systems	Conference	1	0.81%
64.	Iranian Conference on Electrical Engineering	Conference	1	0.81%
65.	Arabian Journal for Science and Engineering	Journal	1	0.81%
66.	IEEE Symposium Series on Computational Intelligence	Conference	1	0.81%
67.	IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference	Conference	1	0.81%
68.	Applied Soft Computing	Journal	1	0.81%
69.	International Conference on Machine Learning and Computing	Conference	1	0.81%
70.	Journal of Information Security and Applications	Journal	2	1.61%

(*continued on next page*)

Table B.8 (*continued*).

No	Publication outlet	Type	#	%
71.	IEEE/IFIP Network Operations and Management Symposium	Symposium	1	0.81%
72.	International Conference on Soft Computing and Pattern Recognition	Conference	1	0.81%
73.	Information Fusion	Journal	1	0.81%
74.	Security and Communication Networks	Journal	1	0.81%
75.	IEICE Transactions on Information and Systems	Journal	1	0.81%
76.	The Computer Journal	Journal	1	0.81%
77.	Journal of Cyber Security Technology	Journal	1	0.81%
78.	Concurrency and Computation: Practice and Experience	Journal	3	2.42%
79.	Security and Privacy	Journal	2	1.61%
80.	Applied Artificial Intelligence	Journal	2	1.61%
81.	Artificial Intelligence and Evolutionary Computations in Engineering Systems	Book chapter	1	0.81%
82.	International Journal of Communication Systems	Journal	2	21.61%
83.	Journal of Ambient Intelligence and Humanized Computing	Journal	1	0.81%
84.	Soft Computing for Problem Solving	Book Chapter	1	0.81%
85.	Information Technology and Mechatronics Engineering Conference	Conference	1	0.81%
86.	ACM Workshop on Wireless Security and Machine Learning	Workshop	1	0.81%
87.	IEEE International Conference on Communications	Conference	1	0.81%
88.	International Conference on Trends in Electronics and Informatics	Conference	1	0.81%
89.	Engineering Applications of Artificial Intelligence	Journal	1	0.81%
90.	Journal of Information and Optimization Sciences	Journal	1	0.81%
91.	International Conference on Computer Science and Engineering	Conference	1	0.81%
92.	Wireless Personal Communications	Journal	1	0.81%

Table C.9
Classification of selected studies w.r.t. detection techniques and other important categories in 2015.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Tama and Rhee [55]	Misuse	Voting	C4.5, SVM, k-NN	CFS	10CV	NSL-KDD	Accuracy: 99.45; precision: 99.4; recall: 99.5; F_1: 99.3
Tama and Rhee [56]	Anomaly	Voting	C4.5, RF, CART	CFS+PSO	10CV	NSL-KDD	Accuracy: 99.0; FPR: 0.2
Sreenath and Udhayan [57]	Misuse	Bagging	NB, DS, HT, ADT	–	10CV	NSL-KDD	Accuracy: 97.85
Robinson and Thomas [125]	Misuse	AD	RF	–	Hold-out	LLS-DDoS, CAIDA 2007, CAIDA Conficker	Accuracy LLS-DDoS: 99.47; CAIDA 2007: 99.89; CAIDA Conficker: 99.96
Gaikwad and Thool [58]	Misuse	Bagging	REPT	Clustering	Hold-out, 10CV	NSL-KDD	Hold-out: accuracy: 81.30; FPR:14.8; 10CV: accuracy: 99.68; FPR: 0.3
Gaikwad and Thool [59]	Anomaly	Bagging	PART	GA	Hold-out, 10CV	NSL-KDD	Hold-out: accuracy: 781.37; FPR:17.2; TPR: 78.4; 10CV: accuracy: 99.72; FPR: 0.3; TPR: 99.7
Choudhury and Bhowal [60]	Anomaly	Boosting, bagging, stacking	NB, LR, k-NN, C4.5, PART, RIPPER, RT, RF, REPT	–	Hold-out	NSL-KDD	Accuracy: 91.52; precision: 95.1; sensitivity: 88.68; specificity: 88.04; AUC: 98.4
Sornsuwit and Jaiyen [27]	Misuse	AB	NB, DT, MLP, SVM, k-NN	CFS	Hold-out	KDD Cup 99	Sensitivity: 76.0; specificity: 99.05
Stevanovic and Pedersen [112]	Anomaly	RF	–	–	10CV	UPC, ISOT, ISCX, MCFP, Contagio, Honeypot	Precision: 99.0; recall: 98.5
Ronao and Cho [131]	Anomaly	Voting	RF	PCA	10CV	Private	FPR: 7.6; FNR: 0.28
Liu et al. [140]	Anomaly	RF	–	–	5CV	KPI	Accuracy: 89.0
Hedar et al. [61]	Anomaly	RF	–	AGAAR	Hold-out	NSL-KDD	Accuracy: 80.67
Elekar [28]	Anomaly	RF	–	–	Hold-out	KDD Cup 99	Accuracy: 92.48
Parhizkar and Abadi [106]	Anomaly	Mean, median, voting, max	OCSVM	–	Hold-out	CSIC 2010	DR: 95.90; FPR: 2.82; accuracy: 96.54
Malik et al. [29]	Misuse	RF	–	PSO	10CV	KDD Cup 99	Accuracy: 96.78; FPR: 0.155

Table C.10
Classification of selected studies w.r.t. detection techniques and other important categories in 2016.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Ponomarev and Atkison [141]	Anomaly	Bagging	REPT	–	–	Telemetry	Accuracy: 92.2
Mehetrey et al. [30]	Anomaly	Bagging	DT	–	Hold-out	KDD Cup 99	Accuracy: 99.47
Medina-Pérez et al. [142]	Misuse	Bagging	Clustering	–	5CV	WUIL	Accuracy: 72.0
Alotaibi and Elleithy [117]	Misuse	Voting	ET, RF, bagging	Extra trees	Hold-out	AWID	Accuracy: 96.32; precision: 96.0; recall: 96.0
Yuan et al. [31]	Misuse	AB	DS	–	Hold-out	KDD Cup 99	FPR: 1.38; precision: 98.63
Ni et al. [32]	Misuse	ET, RF, AB	–	MIC	Hold-out	KDD Cup 99	Accuracy:94.1
Thaseen and Kumar [62]	Misuse	Boosting	SVM	Consistency	10CV	NSL-KDD	Accuracy: 99.0
Tama and Rhee [110]	Anomaly	Rotation forest	20 classifiers	–	5 × 2CV	Wi-Fi intrusion	AUC: 97.72
Rodda and Erothi [63]	Misuse	RF	–	–	10CV	NSL-KDD	Accuracy: 98.9
Rathore et al. [33]	Anomaly	RF, REPT	–	FSR, BER	Hold-out	KDD Cup 99, NSL-KDD	TPR: 99.9; FPR: 0
Mishra et al. [119]	Anomaly	RF	LR	RFE	Bootstrap	Malware (UOC), CAIDA, UNSW-NB	Accuracy: 98.90; FPR: 2.81
Milliken et al. [64]	Anomaly	Stacking	–	–	Hold-out	NSL-KDD, ISCX2012	–
Masarat et al. [34]	Anomaly	RF	–	Gain ratio	Hold-out	KDD Cup 99	Accuracy: 94.4
Mabu et al. [65]	Anomaly	RF	GNP	Random feature selection	Hold-out	NSL-KDD	Accuracy: 83.2; FPR: 25.1
Kulariya et al. [35]	Hybrid	RF, XGBoost	–	–	Hold-out	KDD Cup 99	Accuracy: 91.66; sensitivity: 89.79; specificity: 99.38
Kanakarajan and Muniasamy [66]	Anomaly	RF	–	Information gain, symmetrical uncertainty, CFS	Hold-out, 10CV	NSL-KDD	Accuracy: 85.06; TPR: 85.1; FPR: 12.2; precision: 87.5; F_1: 85.1
Junejo and Goh [132]	Anomaly	RF	–	–	Hold-out	Private	Accuracy: 99.72; precision: 99.70; recall: 99.70; FPR: 4.00
Gupta and Kulariya [36]	Anomaly	RF, XGBoost	–	CFS, chi-squared	–	KDD Cup 99, NSL-KDD	Accuracy: 92.13 sensitivity: 90.85; specificity: 97.43;
Chand et al. [67]	Anomaly	Stacking	SVM, RF	–	10CV	NSL-KDD	Accuracy: 97.50; sensitivity: 93.49; specificity: 98.38; precision: 97.60; recall: 97.60
Ying et al. [37]	Misuse	Voting	BN, RF	Random tree	10CV	KDD Cup 99	AUC: 100
Gaikwad and Thool [68]	Anomaly	AveP	RIDOR, REPT, RT	–	CV, hold-out	NSL-KDD	Accuracy: 99.88; FPR: 0.1; RMSE: 3.51
Lueckenga et al. [69]	Anomaly	Voting	k-NN, DT, AB, SVM	–	Hold-out	NSL-KDD	Accuracy: 99.86; FPR: 11.0; FNR: 15.0
Aburomman and Reaz [38]	Misuse	Voting	SVM	PCA, LDA	Hold-out	KDD Cup 99	Accuracy: 92.162; FPR: 1.96; FNR: 10.85
Aburomman and Reaz [39]	Anomaly	Voting	SVM, k-NN	–	Hold-out	KDD Cup 99	Accuracy: 92.90
Maglaras et al. [133]	Anomaly	Voting	SVM	–	Hold-out	Private	Accuracy: 96.3; FPR: 2.5

Table C.11
Classification of selected studies w.r.t. detection techniques and other important categories in 2017.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Timčenko and Gajin [120]	Anomaly	Bagging, Boosting	–	–	5CV	UNSW-NB15	DR: 100; AUC: 99.9
Miller and Busby-Earle [70]	Anomaly	NB	NB	–	Hold-out	NSL-KDD	Accuracy: 84.14
Kushwaha et al. [40]	Anomaly	RF, AB, Bagging, Stacking	–	Mutual information	Hold-out	KDD Cup 99	Accuracy: 99.89; TPR: 99.0; FPR: 0; precision: 99.0; recall: 99.0; F_1: 99.0
Ajaeiya et al. [134]	Hybrid	Bagging, RF	–	–	Hold-out	Private	Multiclass: TPR: 96.3; FPR:0.9; F_1: 96.2; Anomaly: TPR: 98.34; FPR: 1.6; F_1: 98.34
Vinayakumar et al. [41]	Anomaly	AB, RF	–	–	–	KDD Cup 99, NSL-KDD	Accuracy: 92.7; precision: 99.6; recall: 91.0; F_1: 95.0
Tama and Rhee [111]	Anomaly	Rotation forest, boosting	Conjunctive rule	–	10CV	Wi-Fi intrusion	Accuracy: 92.69; precision: 91.85; FPR: 12.87
Mkuzangwe and Nelwamondo [71]	Anomaly	AB	DS	Information gain	Hold-out	NSL-KDD	Accuracy: 90.0
He et al. [42]	Anomaly	ETC, RF, AB	–	Clustering	–	KDD Cup 99	Accuracy: 94.2
Primartha and Tama [72]	Anomaly	RF	–	N2B	10CV	NSLKDD, UNSW-NB15, GPRS	Accuracy: 99.57; FPR: 0.34
Kumar et al. [135]	Anomaly	Voting, MaxP, PP	C4.5, RF, RIPPER, RIDOR, PART	RFC-5103 BiFlow	10CV	Private	TPR: 100.0; FPR: 2.1; TNR: 97.9; FNR: 0,0; Accuracy: 98.2; AUC: 98.9
Jabbar et al. [108]	Anomaly	AveP	RF, average one dependency estimator	–	10CV	Kyoto 2006+	DR: 90.51; DR: 92.38; FPR: 0.14; Huberts index: 80.0
Belavagi and Muniyal [73]	Misuse	RF	–	DT	Hold-out	NSL-KDD	Average Accuracy: 94
Yousefi-Azar et al. [74]	Anomaly	XGBoost	–	–	Hold-out	NSL-KDD	Accuracy: 83.34
Branitskiy and Kotenko [143]	Anomaly	Voting, stacking, Fix and Hodges method	NN, neuro-fuzzy networks, SVM	PCA	Hold-out, 3CV	DARPA 1998	TPR: 99.78; FPR: 0.46; CCR: 98.46; CCR': 0; GAR: 99.72; OVR: 0.2
Branitskiy and Kotenko [43]	Hybrid	Classification tree	Immune systems, NN, neuro-fuzzy classifier, SVM	PCA	Hold-out	KDD Cup 99, NSL-KDD	FPR: 3.19; TPR: 99.99; CC: 99.29
Ludwig [75]	Anomaly	Voting	Autoencoder, deep belief neural network, DNN, ELM	–	Hold-out	NSL-KDD	Accuracy: 92.49; AUC: 91.62; FPR: 14.72; DR: 97.95; precision: 90.0; recall: 98.0; F_1: 93.69
Jabbar et al. [144]	Anomaly	Voting	ADT, k-NN	–	Hold-out	Gure KDD	DR: 99.8; FPR: 0; Huberts index: 99.8; Rand. index: 99.9; accuracy: 99.93
Kevric et al. [76]	Anomaly	Sum	RT, NBT	–	Hold-out	NSL-KDD	Sensitivity: 83.9; specificity: 96.2; accuracy: 89.24
Bosman et al. [145]	Anomaly	Heuristic	ELM	–	–	GSB, Intel Lab, Indoor WSN	Precision: 97.81; recall: 88.72
Tama and Rhee [77]	Anomaly	Voting	RF, NBT, LMT, REPT	CFS+ES	10CV	NSL-KDD	Accuracy: 99.77

Appendix A. List of abbreviations

10CV Ten-fold Cross-Validation.
3CV Three-fold Cross-Validation.
5 × 2CV Five-times of Two-fold Cross-Validation.
5CV Five-fold Cross-Validation.
6CV Six-fold Cross-Validation.
AB AdaBoost.
ABC Artificial Bee Colony.
ADT Alternating Decision Tree.
AGAAR Accelerated Genetic Algorithm and Rough Set Theory.
AUC Area Under ROC Curve.
AveP Average of Probability.
BA Bat Algorithm.
BER Backward Elimination Ranking.
BN Bayesian Network.
CART Classification And Regression Tree.
CC Correctly Classified Class.
CCA Canonical Correlation Analysis.
CCR Conflict Cases Rate.

Table C.12
Classification of selected studies w.r.t. detection techniques and other important categories in 2018.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Zwane et al. [121]	Anomaly	RF, AB, Bagging	–	–	6CV	UNSW-NB15	TPR: 90.2; FPR: 5.7; AUC: 98.1
Vinutha and Poornima [78]	Misuse	AB, bagging, stacking		SU	10CV	NSL-KDD	Accuracy: 99.89; TPR; 99.9; FPR: 0.1; precision: 99.9; AUC: 100
Vaca and Niyaz [118]	Misuse	Bagging, ET, RF, XGBoost	–	–	10CV	AWID	Accuracy: 99.1
Pham et al. [79]	Anomaly	Bagging	REPT, RT, C4.5	Naive Bayes, gain ratio	Hold-out	NSL-KDD	Accuracy: 84.25; FPR: 2.79
Kaur and Hahn [136]	Anomaly	Bagging	C4.5, BN	–	Hold-out	Private	–
Ghafir et al. [137]	Misuse	Boosting	–	–	10CV	Private	Accuracy: 83.7
Gautam and Doegar [44]	Misuse	Voting	NB, AB, PART	Information gain	–	KDD Cup 99	Accuracy: 99.97; precision: 99.99; recall: 99.98
Dahiya and Srivastava [122]	Misuse	RT	–	CCA, LDA	Hold-out	UNSW-NB15	Accuracy: 92.16; specificity: 80.43; FPR:1.5; precision: 90.6; recall: 90.3, AUC:99.1
Al-Jarrah et al. [80]	Anomaly	AB, bagging, RF	C4.5	–	10CV	NSL-KDD, Kyoto 2006+	Accuracy: 99.62; DR: 99.54; FPR: 0.3; MCC: 99.2
Bansal and Kaur [126]	Hybrid	XGBoost	–	–	Hold-out	CICIDS 2017	Accuracy: 99.54
Al-jawarneh et al. [157]	Anomaly	Voting	C4.5, MP, RT, REPT, AB, DS, NB	Information gain	Hold-out	NSL-KDD	TPR: 99.7; FPR:0.3; accuracy: 99.81
Vigneswaran et al. [45]	Anomaly	AB, RF	–	–	Hold-out	KDD Cup 99	Accuracy: 92.7; precision: 99.9, recall: 91.0, F_1: 95.3
Soheily-Khah et al. [114]	Misuse	RF	–	–	Hold-out	ISCX 2012	Accuracy: 99.9; DR: 99.9; FPR: 0
Injadat et al. [115]	Anomaly	AB	RF	–	Hold-out	ISCX 2012	Accuracy: 99.93; Precision: 99.9; Recall: 99.9; FPR: 0.1
Belouch et al. [123]	Anomaly	RF	–	–	Hold-out	UNSW-NB15	Accuracy: 97.49; sensitivity: 93.53; specificity: 97.75
Ahmad et al. [116]	Anomaly	RF	–	–	Hold-out	ISCX 2012	Average DR: 99.7; average FPR: 1.0
Zhou et al. [81]	Anomaly	GBT	–	–	Hold-out	NSL-KDD, UNSW-NB15	Accuracy: 99.29; precision: 99.29; recall: 99.29
Zhang et al. [82]	Misuse	Voting	Autoencoder, XGBoost	–	Hold-out	NSL-KDD	Precision: 88.14; recall: 96.95; F_1: 91.97
Thaseen et al. [83]	Anomaly	Voting	SVM, NB, LPBoost	Chi-square	Hold-out, CV	NSL-KDD	DR: 99.236
Zaman and Lung [109]	Anomaly	Voting	K-means, Fuzzy C-means, k-NN, NB, SVM, RBF	–	Hold-out	Kyoto 2006+	Precision: 92.0; recall: 95.83; accuracy: 97.54; AUC: 97.41
Jabbar et al. [84]	Misuse	Voting	ADT, NB	–	Hold-out	NSL-KDD	Accuracy: 100; FPR:0; DR: 100
Shen et al. [46]	Misuse	Random subspace	ELM	–	Hold-out	KDD Cup 99, NSL-KDD, Kyoto 2006+	Accuracy: 98.94; DR: 98.37; FPR: 0.32

CFS Correlation-based Feature Selection.
DBSCAN Density-Based Spatial Clustering of Applications with Noise.
DDoS Distributed Denial of Service Attack.
DL Deep Learning.
DNN Deep Neural Network.
DR Detection Rate.
DS Decision Stump.
DT Decision Tree.
ELM Extreme Learning Machine.
ES Evolutionary Search.
ET Extra Tree.

Table C.13
Classification of selected studies w.r.t. detection techniques and other important categories in 2019.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Verma and Ranga [146]	Anomaly	Boosted tree	–	–	Hold-out	RPL-NDDS17	Accuracy: 94.5; AUC: 98
Tama et al. [85]	Anomaly	Voting	Rotation forest, Bagging	CFS+ES	Hold-out	NSL-KDD, UNSW-NB15	Accuracy: 85.8; sensitivity: 86.8; precision: 88
Subudhi and Panigrahi [138]	Anomaly	Stacking	NB, k-NN, rule induction, DT, RBF	–	10CV	Synthetic	Accuracy: 92.17; sensitivity: 90.86; FPR: 3.51
Illy et al. [86]	Hybrid	Voting, bagging, boosting	k-NN, RF, DT	–	Hold-out	NSL-KDD	Accuracy (anomaly): 85.81; Accuracy (misuse): 83.83
Al-Mandhari et al. [47]	Misuse	RF, NB	Bagging, AB	–	10CV, hold-out	KDD Cup 99	Accuracy: 99.94; TPR: 99.9; FPR: 0.1; Precision: 99.9
Mazini et al. [87]	Anomaly	AB	–	ABC	Hold-out	NSL-KDD, ISCX 2012	DR: 99.61; FPR: 1.0; accuracy: 98.90
Jan [48]	Misuse	AB	NN	–	Hold-out	KDD Cup 99	DR:99.3
Li et al. [49]	Misuse	Voting	RF	Bat algorithm	Hold-out	KDD Cup 99	Accuracy: 96.42; FPR: 0.98
Abdul-hammed et al. [127]	Anomaly	RF	–	–	–	CICIDS 2017	Accuracy: 99.99
Tama and Rhee [88]	Anomaly	GBM	–	–	10CV, hold-out	NSL-KDD, UNSW-NB15, Wi-Fi Intrusion	Accuracy: 99.85; FPR: 0.27
Salo et al. [89]	Anomaly	Voting	SVM, k-NN, MLP	Information gain, PCA	Hold-out	ISCX 2012, NSL-KDD, Kyoto 2006+	Accuracy: 99.01; DR: 99.1; FPR: 1.0; precision: 99.1; F_1: 99.2
Moustafa et al. [90]	Anomaly	AB	DT, NB, NN	Correlation coefficient	10CV	UNSW-NB15, NIMS	Accuracy: 99.54; DR: 98.93; FPR: 1.38
Khonde and Ulaga-muthalvi [91]	Anomaly	Voting	NN, DT, k-NN, RF, SVM	Gini index	Hold-out	NSL-KDD	Accuracy: 98.9; FPR: 0.05
Krish-naveni and Prabakaran [113]	Hybrid	Voting	NB, SVM, LR	–	Hold-out	Honeypot	Accuracy: 92.39; FPR: 0.12
Pandey [92]	Hybrid	Voting	DT, LMT, MP, RF, RT, REPTree, AB, DS, Bagging, NB	Information gain	10CV	NSL-KDD	Accuracy: 99.85; FPR: 3.0
Sornsuwit and Jaiyen [50]	Hybrid	AB	k-NN, DT, MLP, SVM, LDA	CFS	Hold-out	UNB-CICT, NSL-KDD, UNSW-NB15, KDD Cup 99	Accuracy: 99.98; FPR: 0.03

ETC Extremely Randomized Trees.
FNR False Negative Rate.
FPR False Positive Rate.
FSR Forward Selection Ranking.
GA Genetic Algorithm.
GAR Generalization Capability Rate.
GAR-F Greedy Randomized Adaptive Search with Annealed Randomness—Forest.
GBM Gradient Boosting Machine.
GBT Gradient Boosting Tree.
GNP Genetic Network Programming.
HT Hoeffding Tree.
IoT Internet of Things.
IoV Internet of Vehicles.
k-NN K-Nearest Neighbor.
KPI Key Performance Indicator.
LDA Linear Discriminant Analysis.
LMT Logistic Model Tree.
LPBoost Linear Programming Boosting.
LR Logistic Regression.
MaxP Maximum Probability.
MCC Matthew's Correlation Coefficient.
MIC Maximal Information coefficient.
MLP Multilayer Perceptron.
MP Meta Pagging.
NB Naive Bayes.
NBT Naive Bayes Tree.
NN Neural Network.
OCSVM One-class Support Vector Machine.
OVR Over-fitting Rate.
PART Partial Decision Tree.
PCA Principle Component Analysis.
PP Product of Probability.
PSO Particle Swarm Optimization.
RBF Radial Basis Function.
RBM Restricted Boltzmann Machine.
REPT Reduce Error Pruning Tree.
RF Random Forest.

Table C.14
Classification of selected studies w.r.t. detection techniques and other important categories in 2020.

Author(s)	Type	Ensemble scheme	Base learner(s)	Feature selection	Validation technique	Dataset	Best results (%)
Abirami et al. [124]	Anomaly	Stacking	RF, NB, SVM	–	Hold-out	UNSW-NB15	Precision: 94.0; recall: 94.0; F_1: 94.0
Bedi et al. [93]	Hybrid	RF, XGBoost	–	–	Hold-out	CIDDS 2017, NSL-KDD	AUC (anomaly): 99.0; recall (misuse): 97.7; precision (misuse): 81.5
Cheng et al. [128]	Anomaly	Stacking	k-NN, SVM, DT, NB	–	Hold-out	DS2OS	Accuracy: 98.15; recall: 93.69; F_1: 95.64
Dash et al. [129]	Anomaly	AB	DT	–	Hold-out	DS2OS	Accuracy: 100
Du and Zhang [51]	Anomaly	AB	DT	–	Hold-out	KDDCup 99	Accuracy: 92.99
Gormez et al. [147]	Hybrid	CatBoost, XGBoost, RF, stacking	–	–	10CV	Digiturk, Labris	DR: 100; FPR: 0
Gupta and Agrawal [148]	Hybrid	RF, AB	DT	–	Hold-out	IoT Botnet	Accuracy: 99.04; F_1: 97.6; recall: 98.6; precision: 96.1
Hariharan et al. [94]	Anomaly	Bagging, logitboost, RF	–	Chi-square	10CV	NSL-KDD	Accuracy: 99.96; Kappa statistic: 99.93
Huan et al. [149]	Anomaly	AB	–	–	5CV	Moore, ISCXVPN 2016	Precision: 92.6; recall: 82.1; FPR: 1.2
Jafarian et al. [139]	Anomaly	Stacking	NN, NB, DL, SVM	Information gain	Hold-out	Private	Accuracy: 99.92; DR: 99.830; FPR: 0.034; precision: 99.98; recall: 99.87; F_1: 99.93
Jiang et al. [95]	Misuse	XGBoost	–	–	Hold-out	NSL-KDD	Precision: 94.0
Karatas et al. [52]	Misuse	RF, AB, GBM	–	–	5CV	KDD Cup 99, NSL-KDD, CICIDS 2017, CICIDS 2018	Accuracy: 99.35
Kaur [96]	Anomaly	AB, stacking	LR, RF	PCA	10CV, hold-out	NSL-KDD, UNSW-NB15	Accuracy: 91.31; FPR: 2.80
Li et al. [107]	Anomaly	Light GBM, CatBoost	–	–	Hold-out	UNSW-NB15, CSIC 2010, Malicious URLs	Accuracy: 99.49, TPR: 99.82; FPR: 1.45
Liu et al. [97]	Anomaly	Bagging, RF, AB, XGBoost	–	–	10CV	NSL-KDD	Accuracy: 97.0; MCC: 90.5; AUC: 99.6
Otoum et al. [98]	Anomaly	Voting	RF, DBSCAN, RBM	–	Hold-out	NSL-KDD	DR: 100; accuracy: 100
Rai [99]	Anomaly	RF, GBM, XGBoost	–	GA	5CV	NSL-KDD	Accuracy: 92.74
Rajadurai and Gandhi [100]	Misuse	Stacking	GBM, RF	–	Hold-out	NSL-KDD	DR: 99.77; recall: 97.75
Shahraki et al. [53]	Anomaly	Real AB, gentle AB, modest AB	DT	–	10CV	UNSW-NB15, TRAbID, NSL-KDD, KDD Cup 99, CICIDS 2017	DR: 98.86
Singh and Singh [130]	Misuse	GBM	–	–	Hold-out	DS2OS	Accuracy: 99.40; precision: 99.00; recall: 99.00; F_1: 99.00
Swami et al. [101]	Anomaly	Voting	CART, MLP, NB, RF, k-NN	–	10CV	NSL-KDD, UNSW-NB15, CICIDS 2017	Accuracy: 99.68; DR: 99.57; precision: 99.74; F_1: 99.66
Tama et al. [102]	Anomaly	Stacking	RF, GBM, XGBoost	–	Hold-out	CSIC 2010, CICIDS 2017, NSL-KDD, UNSW-NB15	Accuracy: 98.82; FPR: 0.74
Uzun and Balli [103]	Misuse	Bagging, RF, LogitBoost	–	–	Hold-out	NSL-KDD	Accuracy: 99.625; error rate: 0.003; F_1: 99.60
Verma and Ranga [104]	Anomaly	GBM, XGBoost, RF, AB, ETC	–	–	10CV, hold-out	CIDDS-001, UNSW-NB15, NSL-KDD	AUC: 98.77; FPR: 0.038
Wei et al. [54]	Anomaly	Bagging, AB	SVM	–	10CV	Kyoto 2006+, KDD Cup 99	Precision: 88.90; recall: 90.60; F_1: 89.80
Zhou et al. [105]	Hybrid	Voting	DT, RF, ForestPA	CFS+BA	10CV, hold-out	NSL-KDD, AWID, CICIDS 2017	Accuracy: 99.89; DR: 99.90; FPR: 0.12

RFE Recursive Feature Elimination.
RIDOR Ripple Down Rule Learner.
RIPPER Repeated Incremental Pruning to Produce Error Reduction.
RMSE Root Mean Square Error.
ROC Receiver Operating Characteristic.
RT Random Tree.
RUSBoost Random Undersampling Boosting.
SDN Software-defined Network.
SMOTE Synthetic Minority Oversampling TEchnique.
SU Symmetrical Uncertainty.
SVM Support Vector Machine.
TF–IDF Term Frequency–Inverse Document Frequency.
TPR True Positive Rate.
VANET Vehicular Ad-hoc Network.
WSN Wireless Sensor Network.
XGBoost eXtreme Gradient Boosting.

Appendix B. Mapping selected studies by publication types

See Table B.8.

Appendix C. Mapping selected studies by intrusion detection techniques

See Tables C.9–C.14.

References

[1] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (2019) http://dx.doi.org/10.1186/s42400-019-0038-7.
[2] P.A.A. Resende, A.C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 1–36, http://dx.doi.org/10.1145/3178582.
[3] A.A. Aburomman, M.B.I. Reaz, A survey of intrusion detection systems based on ensemble and hybrid classifiers, Comput. Secur. 65 (2017) 135–152.
[4] C.C. Fung, M.A. Roumani, K.P. Wong, A proposed study on economic impacts due to cyber attacks in smart grid: A risk based assessment, in: 2013 IEEE Power Energy Society General Meeting, 2013, pp. 1–5.
[5] C.S. Young, Chapter 1 - information security threats and risk, in: C.S. Young (Ed.), Information Security Science, Syngress, 2016, pp. 3–27.
[6] M. Talabis, J. Martin, Information security risk assessment: Data analysis, in: M. Talabis, J. Martin (Eds.), Information Security Risk Assessment Toolkit, Syngress, Boston, 2012, pp. 105–146.
[7] R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, Machine learning models for secure data analytics: A taxonomy and threat model, Comput. Commun. 153 (2020) 406–440.
[8] R. Luh, S. Marschalek, M. Kaiser, H. Janicke, S. Schrittwieser, Semantics-aware detection of targeted attacks: a survey, Journal of Computer Virology and Hacking Techniques 13 (1) (2017) 47–85.
[9] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and systems magazine 6 (3) (2006) 21–45.
[10] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24.
[11] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall/CRC, 2012.
[12] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, 2014.
[13] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[14] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography, Comput. Statist. Data Anal. 53 (12) (2009) 4046–4072.
[15] G. Folino, P. Sabatino, Ensemble based collaborative and distributed intrusion detection systems: A survey, J. Netw. Comput. Appl. 66 (2016) 1–16.
[16] F. Sakiz, S. Sen, A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and iov, Ad Hoc Netw. 61 (2017) 33–50, http://dx.doi.org/10.1016/j.adhoc.2017.03.006.
[17] N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches, Peer-to-Peer Netw. Appl. 12 (2) (2018) 493–501, http://dx.doi.org/10.1007/s12083-017-0630-0.
[18] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2671–2701, http://dx.doi.org/10.1109/comst.2019.2896380.
[19] R. Chapaneri, S. Shah, A comprehensive survey of machine learning-based network intrusion detection, in: Smart Intelligent Computing and Applications, in: Smart Innovation, Systems and Technologies, 2019, pp. 345–356, http://dx.doi.org/10.1007/978-981-13-1921-1_35.
[20] K.A.P. da Costa, J.a.P. Papa, C.O. Lisboa, R. Munoz, V.H.C. de Albuquerque, Internet of things: A survey on machine learning-based intrusion detection approaches, Comput. Netw. 151 (2019) 147–157, http://dx.doi.org/10.1016/j.comnet.2019.01.023.
[21] P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutor. 21 (1) (2019) 686–728, http://dx.doi.org/10.1109/comst.2018.2847722.
[22] N. Moustafa, J. Hu, J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey, J. Netw. Comput. Appl. 128 (2019) 33–55, http://dx.doi.org/10.1016/j.jnca.2018.12.006.
[23] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering., in: International Conference on Evaluation and Assessment in Software Engineering, Vol. 8, 2008, pp. 68–77.
[24] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Inf. Softw. Technol. 64 (2015) 1–18.
[25] B.A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, Vol. 4, CRC press, 2015.
[26] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, L. Cheng, Droiddet: Effective and robust detection of android malware using static analysis along with rotation forest model, Neurocomputing 272 (2018) 638–646, http://dx.doi.org/10.1016/j.neucom.2017.07.030.
[27] P. Sornsuwit, S. Jaiyen, Intrusion detection model based on ensemble learning for u2r and r2l attacks, in: 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), 2015, pp. 354–359, http://dx.doi.org/10.1109/ICITEED.2015.7408971.
[28] K.S. Elekar, Combination of data mining techniques for intrusion detection system, in: 2015 International Conference on Computer, Communication and Control (IC4), IEEE, 2015, pp. 1–5.
[29] A.J. Malik, W. Shahzad, F.A. Khan, Network intrusion detection using hybrid binary PSO and random forests algorithm, Secur. Commun. Netw. 8 (16) (2015) 2646–2660.
[30] P. Mehetrey, B. Shahriari, M. Moh, Collaborative ensemble-learning based intrusion detection systems for clouds, in: 2016 International Conference on Collaboration Technologies and Systems (CTS), 2016, pp. 404–411, http://dx.doi.org/10.1109/CTS.2016.0078.
[31] Y. Yuan, G. Kaklamanos, D. Hogrefe, A novel semi-supervised adaboost technique for network anomaly detection, in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16, 2016, pp. 111–114, http://dx.doi.org/10.1145/2988287.2989177.
[32] X. Ni, D. He, S. Chan, F. Ahmad, Network anomaly detection using unsupervised feature selection and density peak clustering, in: International Conference on Applied Cryptography and Network Security, Springer, 2016, pp. 212–227.
[33] M.M. Rathore, A. Ahmad, A. Paul, Real time intrusion detection system for ultra-high-speed big data environments, J. Supercomput. 72 (9) (2016) 3489–3510, http://dx.doi.org/10.1007/s11227-015-1615-5.
[34] S. Masarat, S. Sharifian, H. Taheri, Modified parallel random forest for intrusion detection systems, J. Supercomput. 72 (6) (2016) 2235–2258, http://dx.doi.org/10.1007/s11227-016-1727-6.
[35] M. Kulariya, P. Saraf, R. Ranjan, G.P. Gupta, Performance analysis of network intrusion detection schemes using apache spark, in: 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1973–1977, http://dx.doi.org/10.1109/ICCSP.2016.7754517.
[36] G.P. Gupta, M. Kulariya, A framework for fast and efficient cyber security network intrusion detection using apache spark, Procedia Comput. Sci. 93 (2016) 824–831, http://dx.doi.org/10.1016/j.procs.2016.07.238.
[37] W. Ying, S. Yongjun, Z. Guidong, Research on intrusion detection model using ensemble learning methods, in: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016, pp. 422–425.
[38] A.A. Aburomman, M.B.I. Reaz, Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection, in: 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 636–640.
[39] A.A. Aburomman, M.B.I. Reaz, A novel SVM-kNN-PSO ensemble method for intrusion detection system, Appl. Soft Comput. 38 (2016) 360–372.
[40] P. Kushwaha, H. Buckchash, B. Raman, Anomaly based intrusion detection using filter based feature selection on KDD-CUP 99, in: TENCON 2017-2017 IEEE Region 10 Conference, IEEE, 2017, pp. 839–844.

[41] R. Vinayakumar, K.P. Soman, P. Poornachandran, Evaluating effectiveness of shallow and deep networks to intrusion detection system, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1282–1289.

[42] D. He, S. Chan, X. Ni, M. Guizani, Software-defined-networking-enabled traffic anomaly detection and mitigation, IEEE Internet Things J. 4 (6) (2017) 1890–1898, http://dx.doi.org/10.1109/jiot.2017.2694702.

[43] A. Branitskiy, I. Kotenko, Hybridization of computational intelligence methods for attack detection in computer networks, J. Comput. Sci. 23 (2017) 145–156, http://dx.doi.org/10.1016/j.jocs.2016.07.010.

[44] R.K.S. Gautam, E.A. Doegar, An ensemble approach for intrusion detection system using machine learning algorithms, in: 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2018, pp. 14–15.

[45] K.R. Vigneswaran, R. Vinayakumar, K. Soman, P. Poornachandran, Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security, in: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2018, pp. 1–6.

[46] Y. Shen, K. Zheng, C. Wu, M. Zhang, X. Niu, Y. Yang, An ensemble method based on selection using bat algorithm for intrusion detection, Comput. J. 61 (4) (2018) 526–538.

[47] I.S. Al-Mandhari, L. Guan, E.A. Edirisinghe, Investigating the effective use of machine learning algorithms in network intruder detection systems, in: Advances in Information and Communication Networks, in: Advances in Intelligent Systems and Computing, 2019, pp. 145–161, http://dx.doi.org/10.1007/978-3-030-03405-4_10.

[48] T. Jan, Ada-boosted locally enhanced probabilistic neural network for IoT intrusion detection, in: Complex, Intelligent, and Software Intensive Systems, in: Advances in Intelligent Systems and Computing, 2019, pp. 583–589, http://dx.doi.org/10.1007/978-3-319-93659-8_52.

[49] J. Li, Z. Zhao, R. Li, H. Zhang, AI-based two-stage intrusion detection for software defined IoT networks, IEEE Internet Things J. 6 (2) (2019) 2093–2102, http://dx.doi.org/10.1109/jiot.2018.2883344.

[50] P. Sornsuwit, S. Jaiyen, A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting, Appl. Artif. Intell. 33 (5) (2019) 462–482.

[51] H. Du, Y. Zhang, Network anomaly detection based on selective ensemble algorithm, J. Supercomput. (2020) 1–22.

[52] G. Karatas, O. Demir, O.K. Sahingoz, Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset, IEEE Access 8 (2020) 32150–32162.

[53] A. Shahraki, M. Abbasi, Ø. Haugen, Boosting algorithms for network intrusion detection: A comparative evaluation of real adaboost, gentle adaboost and modest adaboost, Eng. Appl. Artif. Intell. 94 (2020) 103770.

[54] J. Wei, C. Long, J. Li, J. Zhao, An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing, Concurr. Comput.: Pract. Exper. (2020) e5922.

[55] B.A. Tama, K.H. Rhee, Performance analysis of multiple classifier system in dos attack detection, in: International Workshop on Information Security Applications, Springer, 2015, pp. 339–347.

[56] B.A. Tama, K.H. Rhee, A combination of PSO-based feature selection and tree-based classifiers ensemble for intrusion detection systems, in: Advances in Computer Science and Ubiquitous Computing, Springer, 2015, pp. 489–495.

[57] M. Sreenath, J. Udhayan, Intrusion detection system using bagging ensemble selection, in: 2015 IEEE International Conference on Engineering and Technology (ICETECH), 2015, pp. 1–4, http://dx.doi.org/10.1109/ICETECH.2015.7275015.

[58] D. Gaikwad, R.C. Thool, Intrusion detection system using bagging ensemble method of machine learning, in: 2015 International Conference on Computing Communication Control and Automation, IEEE, 2015, pp. 291–295.

[59] D.P. Gaikwad, R.C. Thool, Intrusion detection system using bagging with partial decision treebase classifier, Procedia Comput. Sci. 49 (2015) 92–98.

[60] S. Choudhury, A. Bhowal, Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection, in: 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015, pp. 89–95.

[61] A.-R. Hedar, M.A. Omer, A.F. Al-Sadek, A.A. Sewisy, Hybrid evolutionary algorithms for data classification in intrusion detection systems, in: 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, 2015, pp. 1–7.

[62] I.S. Thaseen, C.A. Kumar, An integrated intrusion detection model using consistency based feature selection and lpboost, in: 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1–6.

[63] S. Rodda, U.S.R. Erothi, Class imbalance problem in the network intrusion detection systems, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, 2016, pp. 2685–2688.

[64] M. Milliken, Y. Bi, L. Galway, G. Hawe, Multi-objective optimization of base classifiers in stackingc by NSGA-II for intrusion detection, in: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–8.

[65] S. Mabu, S. Gotoh, M. Obayashi, T. Kuremoto, A random-forests-based classifier using class association rules and its application to an intrusion detection system, Artif. Life Robot. 21 (3) (2016) 371–377, http://dx.doi.org/10.1007/s10015-016-0281-x.

[66] N.K. Kanakarajan, K. Muniasamy, Improving the accuracy of intrusion detection using GAR-forest with feature selection, in: Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, in: Advances in Intelligent Systems and Computing, 2016, pp. 539–547, http://dx.doi.org/10.1007/978-81-322-2695-6_45.

[67] N. Chand, P. Mishra, C.R. Krishna, E.S. Pilli, M.C. Govil, A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection, in: 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), 2016, pp. 1–6.

[68] D. Gaikwad, R. Thool, Darensemble: Decision tree and rule learner based ensemble for network intrusion detection system, in: Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1, in: Smart Innovation, Systems and Technologies, 2016, pp. 185–193, http://dx.doi.org/10.1007/978-3-319-30933-0_20.

[69] J. Lueckenga, D. Engel, R. Green, Weighted vote algorithm combination technique for anomaly based smart grid intrusion detection systems, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 2738–2742.

[70] S.T. Miller, C. Busby-Earle, Multi-perspective machine learning a classifier ensemble method for intrusion detection, in: Proceedings of the 2017 International Conference on Machine Learning and Soft Computing - ICMLSC '17, 2017, pp. 7–12, http://dx.doi.org/10.1145/3036290.3036303.

[71] N.N. Mkuzangwe, F. Nelwamondo, Ensemble of classifiers based network intrusion detection system performance bound, in: 2017 4th International Conference on Systems and Informatics (ICSAI), IEEE, 2017, pp. 970–974.

[72] R. Primartha, B.A. Tama, Anomaly detection using random forest: A performance revisited, in: 2017 International Conference on Data and Software Engineering (ICoDSE), 2017, pp. 1–6, http://dx.doi.org/10.1109/ICODSE.2017.8285847.

[73] M.C. Belavagi, B. Muniyal, Multi class machine learning algorithms for intrusion detection - a performance study, in: Security in Computing and Communications, in: Communications in Computer and Information Science, 2017, pp. 170–178, http://dx.doi.org/10.1007/978-981-10-6898-0_14.

[74] M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula, Autoencoder-based feature learning for cyber security applications, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3854–3861.

[75] S.A. Ludwig, Intrusion detection of multiple attack classes using a deep neural net ensemble, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1–7.

[76] J. Kevric, S. Jukic, A. Subasi, An effective combining classifier approach using tree algorithms for network intrusion detection, Neural Comput. Appl. 28 (1) (2017) 1051–1058.

[77] B.A. Tama, K.-H. Rhee, HFSTE: Hybrid feature selections and tree-based classifiers ensemble for intrusion detection system, IEICE Trans. Inf. Syst. 100 (8) (2017) 1729–1737.

[78] H.P. Vinutha, B. Poornima, An ensemble classifier approach on different feature selection methods for intrusion detection, in: Information Systems Design and Intelligent Applications, in: Advances in Intelligent Systems and Computing, 2018, pp. 442–451, http://dx.doi.org/10.1007/978-981-10-7512-4_44.

[79] N.T. Pham, E. Foo, S. Suriadi, H. Jeffrey, H.F.M. Lahza, Improving performance of intrusion detection system using ensemble methods and feature selection, in: Proceedings of the Australasian Computer Science Week Multiconference, Association for Computing Machinery, Brisband, Queensland, Australia, 2018, pp. 1–6.

[80] O.Y. Al-Jarrah, Y. Al-Hammdi, P.D. Yoo, S. Muhaidat, M. Al-Qutayri, Semi-supervised multi-layered clustering model for intrusion detection, Digit. Commun. Netw. 4 (4) (2018) 277–286, http://dx.doi.org/10.1016/j.dcan.2017.09.009.

[81] Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2018, pp. 262–267.

[82] B. Zhang, Y. Yu, J. Li, Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method, in: 2018 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2018, pp. 1–6.

[83] I.S. Thaseen, C.A. Kumar, A. Ahmad, Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers, Arab. J. Sci. Eng. 44 (4) (2018) 3357–3368, http://dx.doi.org/10.1007/s13369-018-3507-5.

[84] M.A. Jabbar, K. Srinivas, S. Sai Satyanarayana Reddy, A novel intelligent ensemble classifier for network intrusion detection system, in: Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016), in: Advances in Intelligent Systems and Computing, 2018, pp. 490–497, http://dx.doi.org/10.1007/978-3-319-60618-7_48.

[85] B.A. Tama, M. Comuzzi, K.-H. Rhee, TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system, IEEE Access 7 (2019) 94497–94507, http://dx.doi.org/10.1109/access.2019.2928048.

[86] P. Illy, G. Kaddoum, C.M. Moreira, K. Kaur, S. Garg, Securing fog-to-things environment using intrusion detection system based on ensemble learning, in: 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1–7.

[87] M. Mazini, B. Shirazi, I. Mahdavi, Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and adaboost algorithms, J. King Saud Univ. - Comput. Inf. Sci. 31 (4) (2019) 541–553, http://dx.doi.org/10.1016/j.jksuci.2018.03.011.

[88] B.A. Tama, K.-H. Rhee, An in-depth experimental study of anomaly detection using gradient boosted machine, Neural Comput. Appl. 31 (4) (2019) 955–965.

[89] F. Salo, A.B. Nassif, A. Essex, Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection, Comput. Netw. 148 (2019) 164–175.

[90] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet Things J. 6 (3) (2019) 4815–4830, http://dx.doi.org/10.1109/jiot.2018.2871719.

[91] S. Khonde, V. Ulagamuthalvi, Ensemble-based semi-supervised learning approach for a distributed intrusion detection system, J. Cyber Secur. Technol. 3 (3) (2019) 163–188.

[92] S.K. Pandey, Design and performance analysis of various feature selection methods for anomaly-based techniques in intrusion detection system, Secur. Priv. 2 (1) (2019) e56.

[93] P. Bedi, N. Gupta, V. Jindal, I-SIamids: an improved siam-IDS for handling class imbalance in network-based intrusion detection systems, Appl. Intell. (2020) 1–19.

[94] R. Hariharan, I. Thaseen, G. Devi, Performance analysis of single-and ensemble-based classifiers for intrusion detection, in: Soft Computing for Problem Solving, Springer, 2020, pp. 759–770.

[95] H. Jiang, Z. He, G. Ye, H. Zhang, Network intrusion detection based on PSO-xgboost model, IEEE Access 8 (2020) 58392–58401.

[96] G. Kaur, A comparison of two hybrid ensemble techniques for network anomaly detection in spark distributed environment, J. Inf. Secur. Appl. 55 (2020) 102601.

[97] J. Liu, B. Kantarci, C. Adams, Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset, in: Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 2020, pp. 25–30.

[98] S. Otoum, B. Kantarci, H.T. Mouftah, A novel ensemble method for advanced intrusion detection in wireless sensor networks, in: Icc 2020-2020 Ieee International Conference on Communications (Icc), IEEE, 2020, pp. 1–6.

[99] A. Rai, Optimizing a new intrusion detection system using ensemble methods and deep neural network, in: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), IEEE, 2020, pp. 527–532.

[100] H. Rajadurai, U.D. Gandhi, A stacked ensemble learning model for intrusion detection in wireless network, Neural Comput. Appl. (2020).

[101] R. Swami, M. Dave, V. Ranga, Voting-based intrusion detection framework for securing software-defined networks, Concurr. Comput.: Pract. Exper. (2020) e5927.

[102] B.A. Tama, L. Nkenyereye, S.R. Islam, K.-S. Kwak, An enhanced anomaly detection in web traffic using a stack of classifier ensemble, IEEE Access 8 (2020) 24120–24134.

[103] B. Uzun, S. Balli, Performance evaluation of machine learning algorithms for detecting abnormal data traffic in computer networks, in: 2020 5th International Conference on Computer Science and Engineering (UBMK), IEEE, 2020, pp. 165–170.

[104] A. Verma, V. Ranga, Machine learning based intrusion detection systems for IoT applications, Wirel. Pers. Commun. 111 (4) (2020) 2287–2310.

[105] Y. Zhou, G. Cheng, S. Jiang, M. Dai, Building an efficient intrusion detection system based on feature selection and ensemble classifier, Comput. Netw. (2020) 107247.

[106] E. Parhizkar, M. Abadi, OC-WAD: A one-class classifier ensemble approach for anomaly detection in web traffic, in: 2015 23rd Iranian Conference on Electrical Engineering, IEEE, 2015, pp. 631–636.

[107] J. Li, H. Zhang, Z. Wei, The weighted word2vec paragraph vectors for anomaly detection over http traffic, IEEE Access 8 (2020) 141787–141798.

[108] M.A. Jabbar, R. Aluvalu, S.S. Reddy S, RFAODE: A novel ensemble intrusion detection system, Procedia Comput. Sci. 115 (2017) 226–234.

[109] M. Zaman, C.-H. Lung, Evaluation of machine learning techniques for network intrusion detection, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–5.

[110] B.A. Tama, K.-H. Rhee, Classifier ensemble design with rotation forest to enhance attack detection of IDS in wireless network, in: 2016 11th Asia Joint Conference on Information Security (AsiaJCIS), 2016, pp. 87–91, http://dx.doi.org/10.1109/AsiaJCIS.2016.13.

[111] B.A. Tama, K.-H. Rhee, A novel anomaly detection method in wireless network using multi-level classifier ensembles, in: Advanced Multimedia and Ubiquitous Engineering, in: Lecture Notes in Electrical Engineering, 2017, pp. 452–458, http://dx.doi.org/10.1007/978-981-10-5041-1_73.

[112] M. Stevanovic, J.M. Pedersen, An analysis of network traffic classification for botnet detection, in: 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), IEEE, 2015, pp. 1–8.

[113] S. Krishnaveni, S. Prabakaran, Ensemble approach for network threat detection and classification on cloud computing, Concurr. Comput.: Pract. Exper. (2019) e5272.

[114] S. Soheily-Khah, P.-F. Marteau, N. Béchet, Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the ISCX dataset, in: 2018 1st International Conference on Data Intelligence and Security (ICDIS), IEEE, 2018, pp. 219–226.

[115] M. Injadat, F. Salo, A.B. Nassif, A. Essex, A. Shami, Bayesian Optimization with machine learning algorithms towards anomaly detection, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–6.

[116] I. Ahmad, M. Basheri, M.J. Iqbal, A. Rahim, Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection, IEEE Access 6 (2018) 33789–33795, http://dx.doi.org/10.1109/access.2018.2841987.

[117] B. Alotaibi, K. Elleithy, A majority voting technique for wireless intrusion detection systems, in: 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2016, pp. 1–6, http://dx.doi.org/10.1109/LISAT.2016.7494133.

[118] F.D. Vaca, Q. Niyaz, An ensemble learning based wi-fi network intrusion detection system (WNIDS), in: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), IEEE, 2018, pp. 1–5.

[119] P. Mishra, E.S. Pilli, V. Varadharajant, U. Tupakula, Nvcloudids: A security architecture to detect intrusions at network and virtualization layer in cloud environment, in: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016, pp. 56–62.

[120] V. Timčenko, S. Gajin, Ensemble classifiers for supervised anomaly based network intrusion detection, in: 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, 2017, pp. 13–19.

[121] S. Zwane, P. Tarwireyi, M. Adigun, Performance analysis of machine learning classifiers for intrusion detection, in: 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), 2018, pp. 1–5.

[122] P. Dahiya, D.K. Srivastava, Network intrusion detection in big dataset using spark, Procedia Comput. Sci. 132 (2018) 253–262.

[123] M. Belouch, S. El Hadaj, M. Idhammad, Performance evaluation of intrusion detection based on machine learning using apache spark, Procedia Comput. Sci. 127 (2018) 1–6.

[124] M. Abirami, U. Yash, S. Singh, Building an ensemble learning based algorithm for improving intrusion detection system, in: Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, 2020, pp. 635–649.

[125] R. Robinson, C. Thomas, Ranking of machine learning algorithms based on the performance in classifying ddos attacks, in: 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), IEEE, 2015, pp. 185–190.

[126] A. Bansal, S. Kaur, Extreme gradient boosting based tuning for classification in intrusion detection systems, in: Advances in Computing and Data Sciences, in: Communications in Computer and Information Science, 2018, pp. 372–380, http://dx.doi.org/10.1007/978-981-13-1810-8_37.

[127] R. Abdulhammed, M. Faezipour, A. Abuzneid, A. AbuMallouh, Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic, IEEE Sens. Lett. 3 (1) (2019) 1–4, http://dx.doi.org/10.1109/lsens.2018.2879990.

[128] Y. Cheng, Y. Xu, H. Zhong, Y. Liu, Leveraging semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in IoT communication, IEEE Internet Things J. (2020).
[129] P.B. Dash, J. Nayak, B. Naik, E. Oram, S.H. Islam, Model based IoT security framework using multiclass adaptive boosting with SMOTE, Secur. Priv. 3 (5) (2020) e112.
[130] K. Singh, N. Singh, An ensemble hyper-tuned model for IoT sensors attacks and anomaly detection, J. Inf. Optim. Sci. (2020) 1–25.
[131] C.A. Ronao, S.-B. Cho, Random forests with weighted voting for anomalous query access detection in relational databases, in: Artificial Intelligence and Soft Computing, in: Lecture Notes in Computer Science, 2015, pp. 36–48, http://dx.doi.org/10.1007/978-3-319-19369-4_4.
[132] K.N. Junejo, J. Goh, Behaviour-based attack detection and classification in cyber physical systems using machine learning, in: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security - CPSS '16, 2016, pp. 34–43, http://dx.doi.org/10.1145/2899015.2899016.
[133] L.A. Maglaras, J. Jiang, T.J. Cruz, Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems, J. Inf. Secur. Appl. 30 (2016) 15–26.
[134] G.A. Ajaeiya, N. Adalian, I.H. Elhajj, A. Kayssi, A. Chehab, Flow-based intrusion detection system for SDN, in: 2017 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2017, pp. 787–793.
[135] S. Kumar, A. Viinikainen, T. Hamalainen, Evaluation of ensemble machine learning methods in mobile threat detection, in: 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, 2017, pp. 261–268.
[136] K.J. Kaur, A. Hahn, Exploring ensemble classifiers for detecting attacks in the smart grids, in: Proceedings of the Fifth Cybersecurity Symposium on - CyberSec '18, 2018, pp. 1–4, http://dx.doi.org/10.1145/3212687.3212873.
[137] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, F.J. Aparicio-Navarro, Detection of advanced persistent threat using machine-learning correlation analysis, Future Gener. Comput. Syst. 89 (2018) 349–359.
[138] S. Subudhi, S. Panigrahi, Application of OPTICS and ensemble learning for database intrusion detection, J. King Saud Univ. - Comput. Inf. Sci. (2019) http://dx.doi.org/10.1016/j.jksuci.2019.05.001.
[139] T. Jafarian, M. Masdari, A. Ghaffari, K. Majidzadeh, Security anomaly detection in software-defined networking based on a prediction technique, Int. J. Commun. Syst. 33 (14) (2020) e4524.
[140] D. Liu, Y. Zhao, H. Xu, Y. Sun, D. Pei, J. Luo, X. Jing, M. Feng, Opprentice, in: Proceedings of the 2015 ACM Conference on Internet Measurement Conference - IMC '15, 2015, pp. 211–224, http://dx.doi.org/10.1145/2815675.2815679.
[141] S. Ponomarev, T. Atkison, Industrial control system network intrusion detection by telemetry analysis, IEEE Trans. Dependable Secure Comput. 13 (2) (2016) 252–260, http://dx.doi.org/10.1109/TDSC.2015.2443793.
[142] M.A. Medina-Pérez, R. Monroy, J.B. Camia, M. Garcí a Borroto, Bagging-tpminer: a classifier ensemble for masquerader detection based on typical objects, Soft Comput. 21 (3) (2016) 557–569, http://dx.doi.org/10.1007/s00500-016-2278-8.
[143] A. Branitskiy, I. Kotenko, Network anomaly detection based on an ensemble of adaptive binary classifiers, in: Computer Network Security, in: Lecture Notes in Computer Science, 2017, pp. 143–157, http://dx.doi.org/10.1007/978-3-319-65127-9_12.
[144] M.A. Jabbar, R. Aluvalu, S.S.S. Reddy, Cluster based ensemble classification for intrusion detection system, in: Proceedings of the 9th International Conference on Machine Learning and Computing, Association for Computing Machinery, Singapore, Singapore, 2017, pp. 253–257.
[145] H.H.W.J. Bosman, G. Iacca, A. Tejada, H.J. Wörtche, A. Liotta, Spatial anomaly detection in sensor networks using neighborhood information, Inf. Fusion 33 (2017) 41–56, http://dx.doi.org/10.1016/j.inffus.2016.04.007.
[146] A. Verma, V. Ranga, ELNIDS: Ensemble learning based network intrusion detection system for RPL based internet of things, in: 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), IEEE, 2019, pp. 1–6.
[147] Y. Gormez, Z. Aydin, R. Karademir, V.C. Gungor, A deep learning approach with Bayesian optimization and ensemble classifiers for detecting denial of service attacks, Int. J. Commun. Syst. (2020) e4401.
[148] A.R. Gupta, J. Agrawal, The multi-demeanor fusion based robust intrusion detection system for anomaly and misuse detection in computer networks, J. Ambient Intell. Humaniz. Comput. (2020) 1–17.
[149] W. Huan, H. Lin, H. Li, Y. Zhou, Y. Wang, Anomaly detection method based on clustering undersampling and ensemble learning, in: 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE, 2020, pp. 980–984.
[150] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[151] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, in: ICML'96, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996, pp. 148–156.
[152] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. (2001) 1189–1232.
[153] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[154] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: A new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.
[155] H.H. Bock, Classification and Related Methods of Data Analysis, Distributors for the USA and Canada, Elsevier Science Pub. Co., 1988.
[156] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259.
[157] S. Aljawarneh, M. Aldwairi, M.B. Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, J. Comput. Sci. 25 (2018) 152–160.
[158] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[159] B.A. Tama, K.-H. Rhee, An extensive empirical evaluation of classifier ensembles for intrusion detection task, Comput. Syst. Sci. Eng. 32 (2) (2017) 149–158.
[160] L. Breiman, Stacked regressions, Mach. Learn. 24 (1) (1996) 49–64.
[161] M.J. Van der Laan, E.C. Polley, A.E. Hubbard, Super learner, Statist. Appl. Genet. Mol. Biol. 6 (1) (2007).
[162] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (Feb) (2012) 281–305.
[163] J. Bergstra, D. Yamins, D.D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in: Proceedings of the 30th International Conference on Machine Learning, JMLR, 2013, pp. 1–9.
[164] K.M. Ting, I.H. Witten, Issues in stacked generalization, J. Artif. Intell. Res. 10 (1999) 271–289.
[165] S. Džeroski, B. Ženko, Is combining classifiers with stacking better than selecting the best one?, Mach. Learn. 54 (3) (2004) 255–273.
[166] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC press, 1984.
[167] N.E. Breslow, Generalized linear models: checking assumptions and strengthening conclusions, Statist. Appl. 8 (1) (1996) 23–41.
[168] J. Quinlan, C4.5: Programs for Machine Learning, Elsevier, 1992.
[169] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[170] J. Abellán, S. Moral, Building classification trees using the total uncertainty criterion, Int. J. Intell. Syst. 18 (12) (2003) 1215–1225.
[171] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009), IEEE, 2009, pp. 1–6.
[172] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Military Communications and Information Systems Conference (MilCIS), 2015, IEEE, 2015, pp. 1–6.
[173] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSP, 2018, pp. 108–116.
[174] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.
[175] B.W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochim. Biophys. Acta (BBA)-Protein Struct. 405 (2) (1975) 442–451.
[176] P. Baldi, S. Brunak, Y. Chauvin, C.A. Andersen, H. Nielsen, Assessing the accuracy of prediction algorithms for classification: an overview, Bioinformatics 16 (5) (2000) 412–424.
[177] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[178] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inform. Sci. 180 (10) (2010) 2044–2064.
[179] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, 2011.
[180] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (200) (1937) 675–701.
[181] H. Finner, On a monotonicity problem in step-down multiple test procedures, J. Amer. Statist. Assoc. 88 (423) (1993) 920–923.
[182] M. Landry, B. Angela, Machine Learning with R and H2O, Mountain View, CA, 2018.